robotron

RVS-Rechnersystem K 1840

Nutzerhandbuch Rechner

VEB Robotron-Vertrieb Berlin

4 77

robotron

RVS-Rechnersystem K 1840

Nutzerhandbuch Rechner

VEB Robotron-Vertrieb Berlin

Diese Unterlage ist Bestandteil der Gesamtdokumentation "RVS-Rechnersystem K 1840".

Weitere Unterlagen siehe Technische Beschreibung K 1840, Systembeschreibung K 1840.

Die vorliegende Dokumentation entspricht dem Stand von 6/88.

Die Ausarbeitung erfolgte durch ein Kollektiv des
VEB Robotron-Elektronik Dresden
Stammbetrieb des VEB Kombinat Robotron.

Herausgeber:

VEB Robotron-Vertrieb Berlin
Mohrenstr. 62
Berlin 1086

Inhaltsverzeichnis Seite

1 . E I N F Ü H R U N G 1-1

2 . B E D I E N U N G 2-1

2.1. ÜBERBLICK 2-1

2.2. BEDIEN- UND ANZEIGEELEMENTE DES CPU-SCHRANKES 2-1
2.2.1. Systembedienfeld 2-1
2.2.2. Stromversorgungsmoduln 2-2
2.2.3. Konsolrechner K 1620 2-3
2.2.4. Folienspeicher 2-3

2.3. BOOTSTRAP-PROZESS 2-3
2.3.1. Bootstrap-Ausdrücke 2-3
2.3.2. Konsolrechner-Bootstrap- und CPU/Speicher-Bootstrap-Funktionen 2-4
2.3.2.1. Konsolrechner-Bootstrap 2-4
2.3.2.2. CPU/Speicher-Bootstrap 2-5
2.3.3. Ausführung eines Bootstrap 2-5
2.3.3.1. Ablauf des Bootstrap bei Kaltstart 2-5
2.3.3.2. Ablauf des Bootstrap bei Warmstart 2-7

2.4. ZUR ARBEIT MIT DEN KONSOLFOLIENSPEICHERN 2-9

3 . N U T Z E R - R E C H N E R - K O M M U N I K A T I O N M I T T E L S D E S K O N S O L S U B S Y S T E M S 3-1

3.1. ALLGEMEINES 3-1

3.2. KOMMUNIKATION ZWISCHEN NUTZER UND PROGRAMM 3-1

3.3. BETRIEBSMODI DES KONSOLSUBSYSTEMS 3-1

3.4. KONSOLFUNKTIONEN 3-2
3.4.1. Programmsteuerung 3-2
3.4.2. Anzeigen und Modifizieren von gespeicherten Daten 3-2
3.4.3. Taktsteuerung 3-2
3.4.4. Konsolsteuerfunktionen 3-3
3.4.4.1. Standardeinstellungen 3-3
3.4.4.2. Zustand 3-3
3.4.4.3. Ketten und Wiederholen von Kommandos 3-4

3.5. KONSOLPROGRAMM 3-4

3.6. ERLÄUTERUNG DER KONSOLKOMMANDOSPRACHE 3-4
3.6.1. Beispiele 3-5
3.6.2. Abkürzungen der Kommandos 3-6

3.7. KONSOLKOMMANDOS 3-7

3.8. KONSOLKOMMANDOS, DIE BEI LAUFENDER CPU AUSGEFÜHRT WERDEN 3-21

3.9. KOMMENTARE IN KOMMANDOS 3-22

3.10. STEUERZEICHEN UND SPEZIALZEICHEN 3-22

3.11. KOMMANDO-QUALIFIER 3-23
3.11.1. Qualifier für den Adreßtyp 3-24
3.11.2. Qualifier für Datenlänge 3-24
3.11.3. Qualifier für Zahlenbasis 3-25
3.11.4. Lokales Überschreiben der Zahlenbasis 3-25
3.11.5. NEXT-Qualifier (/N) 3-25

3.12. KETTEN VON KOMMANDOS 3-26

3.13. FEHLERMELDUNGEN DES KONSOLSUBSYSTEMS 3-27
3.13.1. Syntaktische Fehler 3-27
3.13.2. Kommando-Fehler 3-28
3.13.3. Mikroroutinen-Fehler 3-28
3.13.4. Fehler der CPU K 2810 3-29
3.13.5. Folienspeicher-Fehler 3-30
3.13.6. Kompatibilitäts-Fehler 3-31
3.13.7. Konsolprogramm-Fehler 3-32

4. DIAGNOSE 4-1

4.1. DIAGNOSESYSTEMSTRUKTUR DES RVS K 1840 4-1
4.1.1. Ebenen des Diagnosesystems 4-1
4.1.2. Laden der Diagnoseprogramme und deren Steuerung (Übersicht) 4-2
4.1.3. Zuordnung der Disketten zu den Diagnoseebenen 4-2
4.1.3.1. Konsolebene (Diagnoseebene 5) 4-2
4.1.3.2. Diagnoseebene 4 4-2
4.1.3.3. Diagnoseebenen 3 und 2 4-3

4.2. DIAGNOSE-SUPERVISOR 4-3
4.2.1. Programm- und Testablaufsteuerkommandos 4-3
4.2.2. Scripting 4-12
4.2.3. Abarbeitungssteuerfunktionen 4-13
4.2.4. Debug- und Dienstprogramm-Kommandos 4-17

BILDER

2-1: Systembedienfeld 2-1

T A B E L L E N

2-1: Bedien- und Anzeigeelemente des Systembedienfeldes 2-2
2-2: Inhalt der ersten vier Langworte des Restart-Parameterblocks 2-8
2-3: Halt-Codes 2-8
3-1: Symbole der Konsolkommandosprache 3-4
3-2: Terme der Konsolkommandosprache 3-5
3-3: Symbolische Adressen für DEPOSIT und EXAMINE 3-9
3-4: Steuerzeichen und Spezialzeichen der Konsolkommandosprache 3-22
4-1: Abarbeitungs-Steuer-Flags 4-13

1. E i n f ü h r u n g

Das Nutzerhandbuch Rechner des RVS-Rechnersystems K 1840 enthält eine Zusammenstellung über die Bedienung des Rechners, die Kommunikation zwischen Nutzer und Rechner mittels des Konsolsystems einschließlich der Darstellung der Konsolkommandosprache sowie einen Überblick über das Diagnosesystem und eine Darstellung der Diagnosesupervisorsprache.
Für weitere Informationen zum RVS-Rechnersystem K 1840 stehen die folgenden Dokumentationen zur Verfügung:

Dokumentationsbezeichnung	Dokumentations-Nr.
RVS-Rechnersystem K 1840 Inhaltsverzeichnis Betriebsdokumentation	1.57.550010.1/97
RVS-Architekturhandbuch	1.57.550001.3/78
RVS-Rechnersystem K 1840 Systembeschreibung	1.57.550012.6/61
RVS-Rechnersystem K 1840 Installationshandbuch Teil 1: Aufstellbedingungen	1.57.550013.4/20
RVS-Rechnersystem K 1840 Installationshandbuch Teil 2: Aufstellungsvorschrift	1.57.550014.2/54
RVS-Rechnersystem K 1840 Nutzerhandbuch Peripherie	1.57.550015.0/53

2. Bedienung

2.1. Überblick

In diesem Kapitel werden die Bedien- und Anzeigeelemente am CPU-Schrank K 0681 sowie die Bootstrap-Prozesse beschrieben.

2.2. Bedien- und Anzeigeelemente des CPU-Schrankes

Bei geschlossenen Türen des CPU-Schrankes (SCH1) sind ausschließlich die Bedien- und Anzeigeelemente des Systembedienfeldes (SBF) des RVS K 1840 sichtbar. Der Zugriff zu den anderen Bedien- und Anzeigeelementen ist nur bei besonderen Test- und Wartungsoperationen erforderlich.

2.2.1. Systembedienfeld

Das Systembedienfeld (SBF) ist ein Teil des Konsolsubsystems (KSS). Es besitzt nur wenige Bedien- und Anzeigeelemente (siehe Bild 2-1), da die üblichen Funktionen (wie z. B. EXAMINE, HALT usw.) durch ein Software-Paket unter Nutzung der Konsolkommandosprache realisiert werden. Das SBF ist mit dem Konsolrechner über das Konsolinterface (CIB) verbunden. Die Bedien- und Anzeigeelemente des SBF beschreibt Tabelle 2-1. Das SBF ist an der rechten Vordertür des CPU-Schrankes untergebracht.

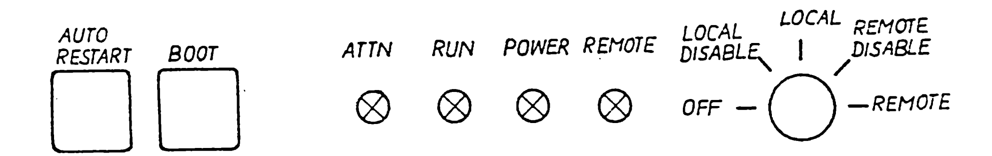


Bild 2-1: Systembedienfeld

2.2.2. Stromversorgungsmoduln

Es gibt fünf Stromversorgungsmoduln (STM) im CPU-Schrank (SCH1). Der STM K 0384 speist den Hauptspeicher, die anderen die übrigen Steckeinheiten, die oberhalb der Stromversorgungsgruppe angeordnet sind. Alle STM bleiben normalerweise eingeschaltet (Schalthebel oben).

Anzeigeelemente (LEDs) auf der Rückseite:

(grün)	Hilfsspannung normal	(rot)	Übertemperatur
(rot)	Überstrom	(rot)	Überstrom
(rot)	Überspannung	(rot)	Überspannung
(grün)	Spannung normal	(grün)	Spannung normal

Anzeigeelemente auf der Frontseite:

(grün) Netz ein

Tabelle 2-1: Bedien- und Anzeigeelemente des Systembedienfeldes

Element	Funktion
AUTO RESTART (rastender Tastenschalter)	Bewirkt in ON-Stellung (untere Position) automatisch einen Systemrestart nach Störung in der Stromversorgung oder nach Fehler-HALT.
BOOT (nichtrastender Tastenschalter)	Das Drücken des Tastenschalters bewirkt das Anfangsladen (Bootstrap) des Betriebssystems. Das Konsolsubsystem geht anschließend in den Programm-E/A-Modus über.
ATTN (Anzeige rot)	Leuchtet, wenn sich die CPU K 2810 im HALT-Zustand befindet.
RUN (Anzeige grün)	Leuchtet, wenn sich die CPU K 2810 im RUN-Zustand befindet.
POWER (Anzeige grün)	Leuchtet, wenn die +5 V anliegen.
REMOTE (Anzeige rot)	Leuchtet, wenn Fern-Terminal-Zugriff möglich.
Drehschalter mit 5 Stellungen:	
OFF	RVS K 1840 ausgeschaltet.
LOCAL DISABLE	Fernzugriff und Konsol-E/A-Modus gesperrt, nur Programm-E/A-Modus möglich.
LOCAL	Fernzugriff gesperrt, Konsol- und Programm-E/A-Modus möglich.

Tabelle 2-1: (Forts.) Bedien- und Anzeigeelemente des Systembedienfeldes

Element	Funktion
REMOTE DISABLE	Fern-Terminal nur als Konsolterminal nutzbar; Nutzung im Konsol-E/A-Modus gesperrt; lokales Terminal wie bei REMOTE.
REMOTE	Fern-Terminal im Konsol- und Programm-E/A-Modus nutzbar; lokales Terminal noch als Konsolterminal nutzbar.

2.2.3. Konsolrechner K 1620

Der Konsolrechner K 1620 befindet sich gewöhnlich im BEREIT-Zustand und ist nur zu bedienen, wenn Wartungsoperationen das erfordern. Bedien- und Anzeigeelemente des Konsolrechners sind auf dessen Frontpaneel-Leiterplatte untergebracht:

- Rastender grüner Tastenschalter HALT/BEREIT mit roter Leuchtdiode für HALT.
- Nichtrastender roter Tastenschalter ANFL-LOE (Anfangslader-Generallöschen) mit roter Leuchtdiode für BEREIT.

2.2.4. Folienspeicher

Die beiden 5.25"-Folienspeicherlaufwerke sind nach Öffnen der Vordertüren des CPU-Schrankes (SCH1) bedienbar.

2.3. Bootstrap-Prozeß

Der Bootstrap-Prozeß des RVS K 1840 erfolgt automatisch, wenn einer der folgenden Zustände im Rechner existiert:

1. Die Konsol- und CPU-Stromversorgung wird bei ON-Stellung des Schalters AUTO RESTART (in ON-Stellung) eingeschaltet.
2. Der K 1840 ist eingeschaltet worden und die BOOT-Taste wird betätigt.
3. Der K 1840 arbeitet und die CPU K 2810 erreicht Time-out wegen ungerader Adresse oder ein unzulässiger Trap-Vektor tritt auf.
4. Der K 1840 arbeitet und entweder die CPU oder der Takt stoppt.

Einige Bootstrap-Operationen erfordern nur wenige, andere gar keine Operatoreingriffe. Nur die von den o.g. Zuständen 1. und 2. ausgehenden Bootstrap-Folgen erfordern Operatoreingriffe.

2.3.1. Bootstrap-Ausdrücke

Zur Beschreibung der Bootstrap-Folgen werden nachfolgend aufgelistete Ausdrücke für Dateien und Programme benutzt:

1. Konsol-Boot:
   Es ist ein in den CIB-ROMs residentes Programm (Anfangslader). Es wird stets dann angesprungen, wenn der K 1840 eingeschaltet wird oder das

Konsolprogramm fehlerhaft abbricht. Dieses Programm lädt das Konsolprogramm von der Diskette DS1 des Konsolfolienspeichers in den RAM des Konsolrechners.

2. Konsolprogramm:
   Es wird vom Konsol-Boot in den RAM des Konsolrechners geladen. Dieses Programm enthält den Konsolkommandosprach-Interpreter, die K 1840-Bootstrap-Funktionen und den Indirektkommandodatei-Prozessor.
3. WCS-Datei:
   Sie ist auf Diskette DS1 enthalten. Sie muß in den ladbaren Mikroprogrammspeicher (WCS) der CPU K 2810 geladen werden, bevor der Befehlssatz der CPU ausgeführt werden kann.
4. RVS K 1840-Primär-Boot-Kommandodatei:
   Diese Datei ist auf Diskette DS1 gespeichert und enthält die Konsolkommandos zum Laden und Starten des Primärladeprogramms. Diese Konsolkommandos werden durch das Konsolprogramm abgearbeitet. Es kann mehrere Primär-Boot-Kommandodateien geben, die als Parameter des Bootkommandos angegeben werden. In der Primär-Boot-Indirektkommandodatei können Laufwerksnummern oder ein Gerät, von dem geladen werden soll, angegeben werden.
5. RVS K 1840-Primärlader:
   Er ist ein Programm auf Niveau des Betriebssystems SVP 1800, das ein Sekundärladeprogramm vom Systemladegerät holt. Die Parameter für das Systemladegerät sind in der jeweiligen Primär-Boot-Indirektkommmandodatei enthalten. Der Primärlader ist auf der Diskette DS1 enthalten, wird in den Hauptspeicher geladen und durch die Primär-Boot-Kommandodatei gestartet.
6. Restart-Routine:
   Diese ist ein ROM-residentes Betriebssystemprogramm, das vom Konsolprogramm während einer automatischen Restartfolge initiiert wird. Sie entscheidet, ob der Restart ein Kalt- oder Warmstart werden soll.
7. Memory-Sizer:
   Ein ROM-residentes Betriebssystemprogramm, das durch die Primär-Boot-Datei während eines Kaltstarts oder einer Boot-Folge initiiert wird. Er lokalisiert einen zusammenhängenden Block von 64 Kbyte im Speicher und sendet die Startadresse zum RVS K 1840-Primär-Boot. Dieses lädt dann den Primärlader in diesen Block.
8. RVS K 1840-Sekundärlader:
   Das ist ein Programm auf Betriebssystemniveau, welches das Betriebssystem SVP 1800 oder den Diagnosesupervisor vom Systemladegerät holt. Der Sekundärlader ist eine Datei auf der Systemplatte oder einem anderen Systemgerät. Er wird durch den Primärlader geladen und gestartet.

2.3.2. Konsolrechner-Bootstrap- und CPU/Speicher-Bootstrap-Funktionen

Im RVS K 1840 gibt es zwei Bootstrap-Stufen:
das Konsolrechner-Bootstrap und das CPU/Speicher-Bootstrap.

2.3.2.1. Konsolrechner-Bootstrap

Das Konsolrechner-Bootstrap umfaßt folgendes:

1. Start eines auf dem CIB-ROM gespeicherten Anfangsladers, wofür es eine Reihe von alternativen Möglichkeiten gibt.

1.1. Startmöglichkeiten, wenn nachfolgend ein CPU-Bootstrap bzw. ein CPU-Restart durchgeführt werden soll.

- automatisch
  beim Netzeinschalten oder
  bei kurzfristigem Netzausfall

- durch Tastendruck: Taste ANF-LOE am Konsolrechner

- aus dem Bedienzustand des K1620 mit
  D oder
  173000G oder
  140000G oder
  140200G
  Die vier Möglichkeiten sind identisch und entsprechen dem Netzeinschalten.

1.2. Startmöglichkeiten, wenn die CPU arbeitet und nicht unterbrochen werden soll (z.B. nach einem Defekt des Konsolrechners oder zum Zwecke des Umlegens der Konsoldiskette auf das andere Floppylaufwerk). Der spezielle Eintrittspunkt bewirkt, daß die Schritte 2 und 5 im folgenden Ablauf übersprungen werden.
   - aus dem Bedienzustand des K1620 mit
     141330G
   - aus dem Konsol-E/A-Modus mit dem Kommando REBOOT

2. Ausführen einer Reihe von Tests des Konsolrechners
3. Laden des Konsolprogramms von Diskette DS1 in den RAM des Konsolrechners
4. Starten des Konsolprogramms
5. Laden des ladbaren Mikroprogrammspeichers (WCS) von Diskette DS1
6. Ist der Schalter AUTO RESTART in ON-Stellung, wird das CPU-Bootstrap initiiert.
7. Ist der Schalter in OFF-Stellung, wird das KSS im Konsol-E/A-Modus gehalten und eine Operatoreingabe erwartet.

2.3.2.2. CPU/Speicher-Bootstrap

Das CPU/Speicher-Bootstrap umfaßt folgendes:

1. In der Stromversorgungs-Einschaltfolge wird das CPU-Initialisierungsmikroprogramm ausgeführt.
2. Die CPU wartet auf den Start einer Konsol-Boot-Folge, der wie folgt ausgelöst werden kann:
   2.1. Eingabe des Konsolkommandos BOOT
   2.2. Betätigen der BOOT-Taste
   2.3. Initiieren einer Auto-Restart-Folge (Schalter AUTO RESTART in ON-Stellung und ein Warm-Restart-Versuch führt zu Fehlern).
3. Wenn eine der vorgenannten Bedingungen auftritt, lädt das Konsolsubsystem ein Bootstrap-Programm in den CPU-Speicher von der Folienspeicherdiskette DS1.
4. Das Konsolsubsystem startet das Primärladeprogramm in der CPU.
5. Das Konsolsubsystem geht in den Programm-E/A-Modus über.

2.3.3. Ausführung eines Bootstrap

2.3.3.1. Ablauf des Bootstrap bei Kaltstart

Von den oben erwähnten Alternativen wird hier die kürzere Möglichkeit mit dem Start durch Netzeinschalten und Setzen der Taste AUTO RESTART angeführt:

1. Konsoldiskette DS1 in das gewünschte Floppy-Laufwerk einlegen (bei Verwendung von Drive 1 darf in Drive 0 währenddessen keine Diskette mit gültigem Boot-Block liegen!)
2. Setzen des Schalters AUTO RESTART in ON-Stellung.
3. Drehschalter in die Stellung LOCAL bringen.
4. Netz einschalten.
5. Nach erfolgtem Konsolrechner-Bootstrap erscheint auf dem Konsolterminal folgende Ausschrift (z.B. liege die Konsoldiskette DS1 in Drive 1 und in Drive 0 liege eine nicht ladbare Diskette, z.B. DS2, DS3 ...):

```
?NO BOOT ON FLOPPY DRIVE_0
BOOTING FROM DRIVE_1

CPU HALTED,SOMM CLEAR,STEP=NONE,CLOCK=NORM
RAD=HEX,ADD=PHYS,DAT=LONG,FILL=00,REL=00000000
DRIVE_0=DIS, DRIVE_1=DX0=CONS
INIT SEQ DONE
HALTED AT 00000000

(RELOADING WCS)
LOAD DONE,0800 MICROWORDS LOADED
VER:PCS=01 WCS=0E-10 FPLA=0E CON=V01-00-L
```

6. Die ON-Stellung des AUTO-RESTART-Schalters initiiert einen RESTART-Versuch, der beim Neueinschalten erfolglos endet und automatisch in einen CPU-Bootstrap übergeleitet wird. Das CPU-Bootstrap wird durch Abarbeiten der Standard-Boot-Kommandodatei DEFBOO.CMD von der Konsoldiskette gesteuert. (Bei Verwendung des Konsolkommandos BOOT können auch andere Boot-Kommandofiles aufgerufen werden.) Bei der Abarbeitung von DEFBOO.CMD werden folgende wesentliche Schritte ausgeführt:

   6.1. Anhalten der CPU K 2810
   6.2. ISB in Grundstellung bringen
   6.3. Initialisieren der CPU K 2810
   6.4. Anlegen der Adresse eines SCB, der im ROM gespeichert ist
   6.5. Laden der allgemeinen Register mit den Daten, die später durch den Primärlader angefordert werden:

R0	Code des Boot-Gerätes
R1	TR# des Boot-Geräteadapters
R2	SKRBUS-Adresse des Gerätesteuerregisters
R3	Nummer des Boot-Gerätes
R4	Boot-Block-Nummer
R5	Software-Boot-Steuerflags
FP	0 (signalisiert dem ROM-Programm, daß keine Maschinenprüfung erwartet wird).

   6.6. Start eines im ROM der CPU gespeicherten Programms mit folgenden Funktionen:
   - Test der Grundfunktionen der CPU K 2810 und Prüfung der Speicherkonfiguration auf Gültigkeit
   - Suche eines zusammenhängenden 64-kByte-Speicherblocks, der keine unkorrigierbaren Fehler enthält
   - Laden der Basisadresse des 64-kByte-Speicherblocks +200H in den Stack-Pointer CSP
   - Setzen des Kaltstart-Flags

6.7. Laden des Primärladers VMB.EXE von der Konsoldiskette in den Hauptspeicher ab der im Stackpointer stehenden Adresse.
6.8. Start des Primärladers..Dieser lädt den Sekundärlader, der wiederum das Betriebssystem lädt. Das Konsolsubsystem geht in den Programm-E/A-Modus über.

2.3.3.2. Ablauf des Bootstrap bei Warmstart

Das Konsolsubsystem verursacht unter folgenden Bedingungen einen Restart der CPU, vorausgesetzt die AUTO-RESTART-Taste ist in ON-Stellung:

- Die Netzspannung liegt nach einem Ausfall wieder an.
- Es liegt eine Halt-Bedingung vor (Halt-Befehl oder Maschinenfehler-Halt)

Das Konsolsubsystem führt dann folgende Schritte für einen Restart der CPU durch:

1. Laden des Konsolprogramms von der Konsoldiskette DS1 (bei Netzausfall werden keine Betriebszustände des K 1620 gerettet).
2. Start des Konsolprogramms wie bei Kaltstart
3. Laden des ladbaren Mikroprogrammspeichers (im Gegensatz zum Hauptspeicher ist der WCS nicht an die Stützbatterie angeschlossen)
4. Testen des Schalters AUTO RESTART:
   - wenn OFF: KSS verbleibt im Konsol-E/A-Modus und fordert mit ">>>" Bedienereingaben.
   - wenn ON : Ausführung der folgenden Schritte:
5. Laden von R12 (AP) mit dem Halt-Code, der die Halt-Ursache identifiziert (siehe hierzu Tabelle 2-3),
   Laden von R10 mit dem Halt-PC,
   Laden von R11 mit dem PSL
6. Abarbeitung der Indirekt-Kommandodatei RESTAR.CMD von der Konsoldiskette DS1, wobei folgende Schritte ausgeführt werden:

   6.1. Absicherung, daß die CPU K 2810 angehalten ist durch Ausführung des Konsolkommandos HALT.
   6.2. Initialisierung der CPU
   6.3. Eintragen der SCB-Adresse in das SCB-Basisregister
   6.4. Löschen unbenutzter Register, Eintragen einer 3 in R1 (TR# des Boot-Geräteadapters)
   6.5. Schreiben einer 0 in das Register FP (signalisiert dem ROM-Programm, daß keine Maschinenprüfung erwartet wird)
   6.6. Start des ROM-Programms (des gleichen ROM-Programms wie beim Kaltstart, aber vom Restart-Eintrittspunkt aus)
   - Das ROM-Programm prüft, ob die RAM-Inhalte gültig sind (d.h. ob sie den Netzausfall überdauert haben)
   - Falls der RAM gültig ist, wird der Speicher nach einem gültigen Restart-Parameter-Block (RPB) durchsucht. Den Aufbau eines RPB zeigt Tabelle 2-2.

Ein gültiger RPB muß folgende Bedingungen erfüllen:
- Beginn an einer Page-Grenze
- Das erste Langwort zeigt auf sich selbst.
- Das zweite Langwort (die Adresse der Restart-Routine) darf nicht Null sein.

Wird kein gültiger RPB gefunden oder das Warmstart-Flag im 4. Wort des RPB ist nicht gesetzt, gibt das ROM-Programm die Steuerung an das Konsolprogramm zurück, das einen Kaltstart initiiert. Ansonsten übergibt das ROM-Programm die Steuerung an die Restart-Routine, deren Adresse im zweiten Langwort des RPB angegeben ist. Die Restart-Routine aktiviert das Betriebssystem an dem Punkt, an dem der Abbruch erfolgt war. Das KSS geht in den Programm-E/A-Modus über.

Tabelle 2-2: Inhalt der ersten vier Langworte des Restart-Parameterblocks

Relativadresse	Inhalt
0	Physische Adresse des RPB
4	Physische Adresse der Restart-Routine
8	Prüfsumme der ersten 31 Langworte der Restart-Routine
C	Warmstart-Flag (Bit 0)

Tabelle 2-3: Halt-Codes

Code	Bedeutung
0	Halt oder Einzelschritt vom Konsolterminal
1	Erfolgreiches T-Kommando
2	CTRL/P wurde auf dem Konsolterminal eingegeben
4	Interrupt-Stack ist nicht gültig oder es ist nicht möglich, den SCB (Systemsteuerblock) zu lesen
5	Doppelbusschreibfehler-Halt
6	Halt-Befehl mit PSL<CM> = 0
7	SCB-Vektor <01:00> = 3
8	SCB-Vektor <01:00> = 2 und kein User-WCS
A	CHMX-Befehl, während mit dem Interrupt-Stack gearbeitet wird
B	CHMX-Befehl und SCB-Vektor <01:00> ungleich 0
11	Stromversorgung ist eingeschaltet, der Restart-Parameter-Block (RPB) wird nicht gefunden, Schalter AUTO RESTART ist in OFF-Stellung
12	Stromversorgung ist eingeschaltet und das Warmstart-Flag ist gelöscht, Schalter AUTO RESTART ist in OFF-Stellung
13	Stromversorgung ist eingeschaltet und es kann kein arbeitsfähiger 64-kByte-Speicherbereich gefunden werden
14	bei eingeschalteter Stromversorgung wird ein Boot-Prozeß versucht, der Boot-ROM ist jedoch defekt oder nicht vorhanden
15	Stromversorgung ist eingeschaltet und das Kaltstart-Flag wurde während der Boot-Subroutine gesetzt
16	Einschalten der Stromversorgung wird gestoppt, Schalter "POWER ON" befindet sich in der Halt-Stellung
FF	Störung im Mikroprogramm-Überprüfungs-Test

2.4. Zur Arbeit mit den Konsolfolienspeichern

Die Konsoldiskette kann auf beiden Laufwerken (LW) betrieben werden. Bei Kaltstart, Warmstart und REBOOT-Kommando werden stets beide Laufwerke nach einer Diskette mit gültigem Boot-Block abgesucht und im Erfolgsfalle Bootstrap ausgeführt. Die Lage der Konsoldiskette wird notiert, so daß auch mit RETURN-Kommandos aus Overlay-Programmen (sowie nach Netzausfall) das CONSOL.SYS wieder richtig rückgeladen wird.
Nach dem Bootstrap des CONSOL.SYS nimmt der Floppy-Driver einen solchen Zustand ein, als gäbe es nur das LW, auf dem die Konsoldiskette liegt. Wenn physisch nur ein LW vorhanden ist, muß ledeglich gesichert werden, daß die zu den Laufwerken gehörende Widerstandskombination auf diesem Laufwerk steckt.
Die Verbindung der physischen Lauferksbezeichnungen (Drive 0/1) mit den logischen Namen (DX 0/1 bzw. CSA 0/1) wird mit dem SHOW-Kommando sowie nach den Kommandos 'ENABLE DUAL' und 'DISABLE DUAL' angezeigt. CSA 0/1 wird nur angegeben,wenn ein Betriebssystem mit dem Boot-Kommando gestartet wurde. Da stets nur CSA 0 oder CSA 1 genutzt wird, wird verallgemeinert CSA x angegeben.
Die Lage der Konsoldiskette DS1 wird durch 'CONS' angezeigt. Wird in der <filespezifikation> kein logischer Gerätename angegeben, so erfolgt der Zugriff

- bei Direktkommandos stets zum Laufwerk mit der Konsoldiskette
- bei Indirektkommandos stets zu dem Laufwerk, von dem das Indirektkommando-File gelesen wird.

Logisch gesperrte Laufwerke werden mit 'DIS' und physisch gesperrte mit 'NO DRIVE X' gekennzeichnet. 'DX 0 oder 'DX 1' können nur angegeben werden, wenn sie einem physischen Laufwerk zugewiesen sind.

Nach dem Bootstrap des RVS K 1840 - Betriebssystems wird automatisch der logische Name CSA x: dem Laufwerk zugewiesen, auf dem die Konsoldiskette nicht liegt (Schutz der Konsoldiskette vor unbefugtem Zugriff durch Systemnutzer). Ein Systemzugriff zur Konsoldiskette ist durch folgende Manipulation möglich (z.B. liege die Konsoldiskette auf Drive 1):

- Rückgang in den Konsol-E/A-Modus mit CTRL/P
- Anzeige des automatisch eingestellten Zustandes mit 'SHOW'

DRIVE 0 = DX 0 = CSA x, DRIVE 1 = DX 1 = CONS (A)

- Sperren des Zugriffs zum zweiten Laufwerk durch das Kommando 'DISABLE DUAL' zwingt auch CSA x auf das Konsollaufwerk:

DRIVE 0 = DIS, DRIVE 1 = DX 1 = CONS = CSA x (B)

- Fortsetzen der Systemarbeit mit Kommando 'SET TERMINAL PROGRAM'

Durch das Kommando 'ENABLE DUAL' kann der zuvor automatisch eingestellte Zustand (A) wiederhergestellt werden.

3. Nutzer-Rechner-Kommunikation mittels des Konsolsubsystems

3.1. Allgemeines

Dieses Kapitel enthält eine Einführung in das Konsolsubsystem und eine detaillierte Beschreibung der Konsolkommandosprache. Das Konsolsubsystem ist das Interface des Nutzers zum Initialisieren, Steuern und zur Fehlersuche im RVS-Rechnersystem K 1840. Das Konsolsubsystem (KSS) besteht aus folgenden Komponenten:

1. Terminal K 8911.80
2. 2 Folienspeicher 5.25"
3. Konsolrechner (K 1620)
4. Konsolinterface (CIB)
5. Systembedienfeld (SBF) am CPU-Schrank

Das wichtigste Bindeglied zwischen dem Nutzer und dem RVS-Rechnersystem K 1840 ist das Terminal K 8911.80 (Konsolterminal oder Nutzerterminal). Zum Konsolsubsystem gehört ein Software-Interface-Paket (Konsolprogramm). Es enthält die Mittel zur Kommunikation zwischen dem Nutzer (am Nutzerterminal) und dem Konsolsubsystem sowie zwischen dem Konsolsubsystem und der CPU K 2810.

3.2. Kommunikation zwischen Nutzer und Programm

Die Kommunikation zwischen dem Nutzer und dem Konsolsubsystem erfolgt durch eine der Software entsprechenden Sprache (z.B. Konsolkommandosprache oder Diagnose-Supervisorsprache). Die Konsolkommandosprache wird in diesem Kapitel beschrieben.

3.3. Betriebsmodi des Konsolsubsystems

Das Konsolsubsystem kann in zwei Betriebsmodi arbeiten: Konsol-E/A-Modus oder Programm-E/A-Modus.
Im Konsol-E/A-Modus werden Eingaben des Nutzers am Terminal (z.B. Konsolkommandos) durch das Konsolprogramm des Konsolrechners interpretiert. Das erkannte Kommando wird ausgeführt (z.B. BOOT, HALT). Das Konsolprogramm wird in den Speicher des Konsolrechners von einer Diskette während des Bootstrap-Prozesses geladen.
Im Programm-E/A-Modus werden Nutzereingaben vom Terminal zur CPU übertragen (zeichenweise). Die Daten werden von der Software des RVS-Rechnersystems K 1840 (Betriebssystem) verarbeitet. Umgekehrt kann die CPU Zeichen zum Terminal bzw. zur Diskette übertragen.

3.4. Konsolfunktionen

Mittels des Konsolsubsystems können folgende Funktionen ausgelöst werden: Programmsteuerung, Anzeigen und Modifizieren von gespeicherten Daten, Taktsteuerung, Konsolsteuerfunktionen.

3.4.1. Programmsteuerung

Das Konsolsubsystem wird benutzt, um die CPU zu initialisieren (INITIALIZE-Kommando der Konsolkommandosprache). Es kann dazu verwendet werden, die Arbeit des Betriebssystems einzuleiten und zu beenden. Es wird für den Bootstrap-Prozeß des RVS-Rechnersystems K 1840 benutzt. Dieser Bootstrap-Prozeß führt das Laden des Hauptspeichers des Rechnersystems mit einem Bootstrap-Programm von der Konsoldiskette und Starten des Programms durch.

3.4.2. Anzeigen und Modifizieren von gespeicherten Daten

Das Konsolsubsystem kann dazu benutzt werden, Daten (Speicherplätze) im Hauptspeicher, in E/A-Registern, allgemeinen Registern und internen Registern sowie im Speicher des Konsolrechners anzuzeigen und zu modifizieren.
Auf die Speicherplätze kann in folgenden "Portionen" zugegriffen werden:

1. Hauptspeicher und Speicher des Konsolrechners:
   Byte, Wort, Langwort, Quadwort
2. allgemeine Register (R0-11, AP, FP, SP, PC) und Prozessorregister:
   Byte, Wort, Langwort
3. E/A-Register: Byte, Wort, Langwort (abhängig von der Registerlänge)

3.4.3. Taktsteuerung

Die CPU-Taktrate kann vom Konsolsubsystem (KSS) gesteuert werden. Das ist nützlich für die Fehlersuche in Hardware und Software.
Es gibt drei Konsolsteuermodi der Taktrate:

1. Einzelbefehlsschritt
2. Einzel-ISB-Buszyklusschritt
3. Einzel-ISB-Taktphasenschritt

Im Einzelbefehlsschritt-Modus führt das Betriebssystem einen einzigen Befehl aus und geht danach in den Halt-Zustand.
Im Einzel-ISB-Buszyklusschritt-Modus wird die CPU gestoppt, wenn die ISB-Taktphase 1 (T1) erreicht (gesetzt) wird. T1 bleibt gesetzt bis ein weiterer Zyklus über das KSS eingeleitet wird.
Im Einzel-ISB-Taktphasenschritt-Modus verbleibt die CPU in einer der Taktphasen (T1, T2, T3, T4), bis ein Steuersignal vom KSS empfangen wird. Dann geht die CPU zur nächsten Taktphase über und verbleibt dort.

3.4.4. Konsolsteuerfunktionen

Es gibt folgende Konsolsteuerfunktionen:

1. Steuern der numerischen Zahlenbasis von Konsoleingaben
2. Steuern der Konsoladressierungsmodi
3. Steuern der Konsoldatenlänge
4. Anzeige des Zustands von KSS und CPU
5. Wiederholen von Kommandos
6. Ketten von Kommandos

3.4.4.1. Standardeinstellungen

Vom Konsolsubsystem können Standards für die Adressierungsmodi eingestellt sowie die Zahlenbasis numerischer Ein- und Ausgaben und die Datenlänge im adressierbaren Speicher definiert werden. Die Standardeinstellungen erfolgen durch das SET DEFAULT-Kommando. Standardeinstellungen können innerhalb eines Kommandos durch Qualifier überschrieben oder durch das SET DEFAULT-Kommando geändert werden.

1. Standardadressierungsmodi:

   virtuell (Hauptspeicher)
   physisch (Hauptspeicher)
   Register (allgemein oder intern)
   Speicher des Konsolrechners

2. Standard-Zahlenbasis-Einstellungen für numerische Ein- und Ausgaben:

   oktal
   hexadezimal

3. Standarddatenlängen für den adressierbaren Speicher:

   Byte
   Wort
   Langwort
   Quadwort

4. Standardeinstellungen nach dem Netzeinschalten:

   Zahlenbasis = hexadezimal
   Datenlänge = Langwort
   Adressierung = physisch

3.4.4.2. Zustand

Der Zustand des Konsolsubsystems und der CPU-Komponenten kann auf dem Terminal angezeigt werden. Der CPU-Zustand umfaßt den HALT/RUN - Zustand und den augenblicklichen Taktschritt-Modus. Der Konsolzustand umfaßt die augenblicklichen Standardeinstellungen, den Konsolstatus sowie die Anzahl der Terminal-Füllzeichen.
Der Zustand wird bei jedem Boot-Ablauf des Konsolprogramms, jeder Rückkehr ins Konsolprogramm sowie jedem SHOW-Kommando der Konsolkommandosprache angezeigt.

3.4.4.3. Ketten und Wiederholen von Kommandos

Eine Folge von Konsolkommandos kann zu einer einzigen ausführbaren Liste gekettet werden. Die Kommandos in der Liste können einmalig oder wiederholt (unter Nutzersteuerung) ausgeführt werden. Einzelne und gekettete Kommandos können wiederholt werden. Das Wiederholen wird solange ausgeführt, bis es vom Nutzer abgebrochen wird.
Diese Eigenschaften sind Hilfen bei der Fehlersuche. Siehe dazu die Konsolkommandos LINK, REPEAT, PERFORM sowie Abschnitt 3.12.

3.5. Konsolprogramm

Das Konsolprogramm besteht aus einer Reihe von Moduln. Es ist im 4-Kworte-ROM und 8-Kworte-RAM des Konsolrechners gespeichert. Das Konsolprogramm enthält Moduln zur Konsolsteuerung, zum Nutzerinterface, zur Mikrodiagnose sowie Konsolhilfsfunktionen. Das Konsolprogramm ist die meiste Zeit in der Null- oder Warteschleife. Die Nullschleife enthält eine Reihe von Tests und bedingten Verzweigungen. Wenn eine der Verzweigungsbedingungen erfüllt ist (das entsprechende Flag ist gesetzt), verzweigt das Programm zur entsprechenden Routine.

3.6. Erläuterung der Konsolkommandosprache

Zur Beschreibung der Konsolkommandosprache werden die in Tabelle 3-1 aufgeführten Symbole verwendet, die nicht Bestandteil der Kommandosprache sind und nicht eingegeben werden.

Tabelle 3-1: Symbole der Konsolkommandosprache

Symbol	Bedeutung
()	kennzeichnet eine Kommandoabkürzung
[]	kennzeichnet eine wahlweise Angabe, z.B. NEXT[<blank><count>] zeigt an, daß das count-Argument im NEXT-Kommando wahlweise angegeben werden kann
<>	allgemeines Kennzeichen für ein beschreibendes Symbol, z.B. gibt <address> an, daß eine gültige Adresse im Kommando eingefügt werden muß
<blank>	ein oder mehrere Leerzeichen und/oder Tabulatoren
<count>	numerisches Zählerargument
<XYZ-list>	zeigt an, daß eine oder mehrere Einheiten der Kategorie XYZ angegeben werden sollen
<address>	Adresse
<data>	numerisches Argument

Tabelle 3-1: (Forts.) Symbole der Konsolkommandosprache

Symbol	Bedeutung
<qualifier>	Qualifier (dient der näheren Spezifikation der Kommando-ausführung)
<CR>	"carriage return" des Konsolterminals; wird zum Beenden jeder Kommandozeile verwendet
<LF>	"line feed" des Konsolterminals

In Tabelle 3-2 sind die Terme, die Bestandteile der Syntax der Konsolkommando-sprache sind, aufgeführt.

Tabelle 3-2: Terme der Konsolkommandosprache

Term	Bedeutung
/	Trennt ein Kommando von seinen Qualifiers
	Trennzeichen in Listen
	Trennzeichen in einer Kommandozeile
!	in einer Konsolkommandozeile: leitet Kommentare ein
?	Leitet Fehlermeldungen des Konsolprogramms ein
>>>	Eingabeanforderung (input prompt) an Nutzer im Konsolkommando-sprachniveau
<<<	Eingabeanforderung im Link-Modus
DS>	Eingabeanforderung im Diagnose-Supervisor-Kommandosprachniveau
MIC>	Eingabeanforderung des Mikrotestmonitors
$	Eingabeanforderung des RVS-Betriebssystems SVP 1800

3.6.1. Beispiele

Im folgenden sind Beispiele für eine Kommandozeile angegeben.

EXAMINE[<qualifier-list>][<blank><address>]<CR>

Für das EXAMINE-Kommando gibt es zwei Optionen:

1. Eine Liste von einem oder mehreren Qualifiern
2. Ein Adreß-Argument. Wenn eine Adresse angegeben wird, muß sie durch ein oder mehrere Leerzeichen oder Tabulatoren abgetrennt werden.

Die folgenden Kommandos sind zulässig:

Kommandozeile	Wirkung
EXAMINE	Keine Qualifier angegeben. Es werden die Standardeinstellungen (durch SET DEFAULT) verwendet.
EXAMINE/BYTE/VIRTUAL	Adresse gemäß Standardeinstellung
EXAMINE<blank>1234	keine Qualifier angegeben (Standardeinstellungen), aber ein Speicheradreßargument ist angegeben.
EXAMINE/WORD<blank>1234	Wort-Qualifier angegeben, als Adreß-Qualifier wird Standardeinstellung verwendet.

3.6.2. Abkürzungen der Kommandos

Konsolkommandowörter können abgekürzt werden. Die minimale Abkürzung jedes Kommandowortes ist in Klammern in der Beschreibung jedes Konsolkommandos angegeben.

Beispiel:

```
>>>EXAMINE/VIRTUAL/BYTE 1234
```

kann abgekürzt werden zu:

```
>>>E/V/B 1234
```

Wenn ein inkorrektes Kommando eingegeben wird, reagiert das Konsolprogramm mit der Wiederholung des Kommandos und einer Fehlermeldung.

Beispiel:

```
>>>HLP

        ?'HLP'is incorrect.
```

BEMERKUNG: Das Konsolkommando HELP ABBREV liefert eine Liste von im System verwendbaren Abkürzungen.

3.7. Konsolkommandos

Die Konsolkommandos sind englischsprachige Eingaben oder ihre zugelassenen Abkürzungen, die vom Nutzer auf dem Konsolterminal eingegeben werden. Die Kommandos können nur eingegeben werden, wenn sich das KSS im Konsol-E/A-Modus befindet. Durch ein gültiges Kommando wird die zugehörige Folge von Operationen gestartet. Ist diese Folge abgearbeitet, kann das nächste Konsolkommando eingegeben werden.
Im folgenden werden die Konsolkommandos beschrieben.

BOOT (B)

Syntax: BOOT<blank>[<device-name>][<mnemonic>]<CR>

Das BOOT-Kommando initiiert eine RVS K1840-Systembootstrap-Folge. Das Kommando kann Bootstrap-Operationen von oder für eine Reihe von RVS K1840-Systemgeräten unterstützen. Der Prozeß wird durch auf die Systemgeräte zugeschnittene Indirekt-Kommandodateien gesteuert.

<device-name> ist dabei der logische Name des Floppy-Laufwerks, von dem diese Indirekt-Kommandodatei gelesen werden soll. Wird <device-name> nicht angegeben, erfolgt Zugriff zur Konsoldiskette.

<mnemonic> definiert den Namen der Indirekt-Kommandodatei und besteht aus drei alphanumerischen Zeichen:
DD = Gerätemnemonik aus zwei Buchstaben
n = einstellige Nummer der Geräteeinheit.
Es wird damit die Indirekt-Kommandodatei DDn.BOO.CMD gerufen. Wird <mnemonic> nicht angegeben, wird DEFBOO.CMD gerufen. Wird n nicht angegeben, so wird 0 angenommen.

Mit der Kurzform 'B<CR>' wird DEFBOO.CMD von der Konsoldiskette gerufen. Ein Bootstrap von Geräten, die nicht zu den Systemstandardgeräten gehören, wird mittels einer indirekten Kommandodatei, die die notwendigen Konsolkommandos für die Boot-Folge vom angegebenen Gerät enthält, ausgeführt.
Nach Ausführen des BOOT-Kommandos ist der Zugriff des SVP 1800 zum Konsol-Floppy-Laufwerk mit der Konsoldiskette (DS1) gesperrt, während das jeweils andere der beiden Konsol-Floppy-Laufwerke über den Namen CSA 1 angesprochen werden kann.

Beispiel:

```
>>>B DU0

        CPU HALTED
        INIT SEQ DONE
        HALT INST EXECUTED
        HALTED AT xxxxxxxx

        G 0000000E 00000200
        LOAD DONE, 00005000 BYTES LOADED
          .
          .
```

CLEAR SOMM (CL SO)

Syntax: CLEAR SOMM<CR>

Die SOMM- (STOP ON MICROBREAK MATCH)-Erlaubnis auf dem Konsolinterface wird gelöscht (vgl. Kommando SET SOMM).

CLEAR STEP (CL S)

Syntax: CLEAR STEP<CR>

Ein eingestellter Taktschritt-Modus wird gelöscht; die CPU arbeitet im normalen (free running) Modus (vgl. Kommando SET STEP).

CONTINUE (C)

Syntax: CONTINUE<CR>

Dieses Kommando veranlaßt die CPU, die Programmausführung an der im CPU-Befehlszähler (PC) enthaltenen Adresse zu beginnen. Die CPU wird nicht initialisiert. Das Konsolsubsystem geht nach Ausführung des CONTINUE-Kommandos in den Programm-E/A-Modus über.

DEPOSIT (D)

Syntax: DEPOSIT[<qualifier list>]<blank><address><blank><data><CR>

Qualifier: /BYTE (B), /WORD (W), /LONG (L), /QUAD (Q), /NEXT (N), /VIRTUAL (V), /PHYSICAL (P), /IDBUS (ID), /VBUS (VB), /INTERNAL (I), /GENERAL (G), /CONSOLE (CON), /HEX (H), /OCT (O)

Das Kommando schreibt <data> in die spezifizierte <address> ein. Der benutzte Adreßraum hängt von den Bedingungen (Qualifiers) ab, die im Kommando spezifiziert wurden. Wurden keine Qualifiers angegeben, werden die Daten- und Adressenstandards (aus Kommando SET DEFAULT) zur Festlegung des Adreßraumes benutzt.
Tabelle 3-3 enthält symbolische Adressen, die von den Kommandos DEPOSIT und EXAMINE benutzt werden.

Tabelle 3-3: Symbolische Adressen für DEPOSIT und EXAMINE

Symbol	Definition
PSL	Schreibt oder liest das Prozessorstatus-Langwort
Rn	Schreibt oder liest das allgemeine Register (n=0,1,...,15)
PC	Schreibt oder liest den Befehlszähler
SP	Schreibt oder liest den Stack-Pointer
FP	Schreibt oder liest den Frame-Pointer
AP	Schreibt oder liest den Argument-Pointer
+	Schreibt oder liest den Speicherplatz, der unmittelbar dem zuletzt angesprochenen folgt. Für physische und virtuelle Hauptspeicher-adressen ergibt er sich aus der letzten Adresse plus n, wobei gilt: n = 1 für Bytes n = 2 für Worte n = 4 für Langworte n = 8 für Quadworte Für alle anderen Adreßräume ist n = 1 .
-	Schreibt oder liest den Speicherplatz, der unmittelbar vor dem zu-letzt angesprochenen Speicherplatz liegt.
*	Schreibt oder liest die zuletzt angesprochenen Speicherplätze.
@	Verwendet das zuletzt geschriebene oder gelesene Datum als Adresse.

Beispiel:

```
>>>D/Q/P/O 13777 1777777777777777777777

>>>
```

DISABLE DUAL (DIS DU)

Syntax: DISABLE DUAL<CR>

Dieses Kommando verhindert den Zugriff des Konsolprogramms bzw. des SVP 1800 zum zweiten Konsol-Floppy-Laufwerk. Nach Ausführen des Kommandos wird der Zustand des Floppy-Drivers automatisch angezeigt:

1. wenn die Konsoldiskette auf Laufwerk 0 liegt:

```
DRIVE_0=DX0=CONS[=CSAx]        DRIVE_1=DIS
bzw.
DRIVE_0=DX0=CONS[=CSAx]        NO DRIVE_1
```

2. wenn die Konsoldiskette auf Laufwerk 1 liegt:

DRIVE_0=DIS, DRIVE_1=DX0=CONS[=CSAx]
bzw.
NO DRIVE_0, DRIVE_1=DX0=CONS[=CSAx]

Dieser Zustand wird nach dem Boot- bzw. Reboot-Ablauf des Konsolprogramms automatisch eingenommen und mit dem SHOW-Kommando angezeigt. Die Anzeige von CSAx erfolgt erst nach Systemstart.

DIRECTORY (DIR)

Syntax: DIRECTORY[<blank><device name>]<CR>

Es wird die aktuelle Directory der Diskette auf dem angegebenen Laufwerk auf dem Konsolterminal angezeigt. Wird <device-name> nicht angegeben, wird die Directory der Konsoldiskette angezeigt.

DUPLICATE (DUP)

Syntax: DUPLICATE <CR>

Duplizieren einer vollständigen Diskette.
Als zusätzliche Parameter werden bei Kommandoausführung die Nummern von Ein- und Ausgabelaufwerk angefordert. Es werden RT-11-Disketten vorausgesetzt, die mit Offset = 1 formatiert sind. Damit wird eine Duplizierzeit von 90 s erreicht, bei größeren Offsets ist die Zeit geringfügig und bei Offset = 0 wesentlich länger.

ENABLE DUAL (EN DU)

Syntax: ENABLE DUAL<CR>

Dieses Kommando gibt den Zugriff zum zweiten Floppy-Laufwerk frei, falls dieses installiert ist, d.h. physisch ein System mit 'Dual Drive' vorliegt. Nach Ausführung des Kommandos wird der Zustand des Floppy-Drivers automatisch angezeigt:

1. Konsoldiskette liegt auf Laufwerk 0

   DRIVE_0=DX0=CONS, DRIVE_1=DX1[=CSAx]
   bzw.
   DRIVE_0=DX0=CONS[=CSAx] NO DRIVE_1

2. Konsoldiskette liegt auf Laufwerk 1

   DRIVE_0=DX0[=CSAx] DRIVE_1=DX1=(CONS)
   bzw.
   NO DRIVE_0, DRIVE_1=DX1=(CONS)[=CSAx]

Die Anzeige von CSAx erfolgt erst nach Systemstart mit dem BOOT-Kommando. Im Bootprozeß wird dieses Kommando impliziert durchgeführt, d.h. der Floppy-Driver des SVP 1800 greift z.B. über den Namen CSA 1: stets zu dem Laufwerk zu, auf dem

die Konsoldiskette n i c h t liegt (falls zwei Laufwerke physisch vorhanden sind). Eine Veränderung dieses Standards ist durch Rückgang in den Konsolmodus mit den Kommandos EN DUAL bzw. DIS DUAL möglich. Bei laufendem SVP wird mit dem SHOW-Kommando zusätzlich noch angezeigt, welchem physischen Laufwerk der logische Name CSA 1: zugeordnet ist. Diese Anzeige wird durch das HALT unterdrückt und durch CONTINUE wieder aktiviert.

EXAMINE (E)

Syntax: EXAMINE[<qualifier list>][<blank><address>]<CR>

Qualifier: /BYTE (B), /WORD (W), /LONG (L), /QUAD (Q), /NEXT (N), /VIRTUAL (V), /PHYSICAL (P), /IDBUS (ID), /VBUS (VB), /INTERNAL (I), /GENERAL (G), /CONSOLE (CON), /HEX (H), /OCT (O)

Das EXAMINE-Kommando liest und zeigt den Inhalt von <address> an. Ist <address> nicht angegeben, so wird der Inhalt der Adresse angezeigt, die durch den Adreß-Standard (durch SET DEFAULT-Kommando) definiert wird. Das <address>-Argument kann auch eine symbolische Adresse sein (vgl. DEPOSIT).

Beispiel:

```
>>>E/Q/P/O 13777

        P 13777 1777777777777777777777
>>>
```

EXAMINE IR (E IR)

Syntax: EXAMINE IR<CR>

Zeigt das Befehlsregister der CPU an.

Beispiel:

```
>>>E IR

        IR 11 FE 00
>>>
```

FORMAT (FO)

Syntax: FORMAT <CR>

Formatieren einer Diskette.
Als zusätzliche Parameter werden bei Kommandoausführung angefordert:

Offset: Parameter zur Bestimmung der Reihenfolge der Sektoren innerhalb einer Spur
Es sind zu verwenden:
Für DS0 (LPROEM): Offset = A
Für DS1...DSx (RT-11): Offset = 1

To Drive: Physische Laufwerksnummer.

Das Ende des eigentlichen Formatierens wird angezeigt. Die nachfolgende Prüfung der Diskette durch Schreiben und Kontrollesen eines Testmusters kann ggf. mit CTRL/C abgebrochen werden.

HALT (H)

Syntax: HALT<CR>

Das HALT-Kommando stoppt die CPU nach Beendigung des gerade bearbeiteten Befehls.

Beispiel:

```
>>>H
        HALTED AT xxxxxxxx
>>>H
        CPU HALTED
>>>
```

HELP (HE)

Syntax: HELP<CR>

Das Konsolprogramm ruft eine indirekte Kommandodatei (help file) auf der Konsoldiskette auf und gibt ihren Inhalt aus. Diese Datei enthält eine Beschreibung aller Konsolkommandos.

HELP ABBREV (HE ABBREV)

Syntax: HELP ABBREV<CR>

Das Konsolprogramm ruft eine indirekte Kommandodatei (abbreviation help file) auf und gibt ihren Inhalt aus. Diese Datei enthält eine Liste von Abkürzungen und Regeln für die Konsolkommandosprache.

HELP ERROR (HE ERR)

Syntax: HELP ERROR<CR>

Das Konsolprogramm ruft eine indirekte Kommandodatei (error help file) auf und gibt den Inhalt aus. Diese Datei enthält eine Liste aller Fehlermitteilungen für die Konsolkommandosprache.

INDIRECT (@)

Syntax: @<filename><CR>

Dieses Kommando veranlaßt das Konsolprogramm, die durch <filename> spezifizierte Datei zu eröffnen und die Ausführung der darin enthaltenen Konsolkommandos zu beginnen und solange fortzusetzen, bis eines der folgenden Ereignisse eintritt:

a. Ein WAIT DONE-Kommando wird von der Datei gelesen.
b. Das Ende der indirekten Datei ist erreicht: das Konsolprogramm gibt <@EOF> und '>>>' aus.
c. Der Nutzer gibt ein CTRL/C ein; die Abarbeitung der Kommandodatei wird abgebrochen.

INITIALIZE (I)

Syntax: INITIALIZE<CR>

Das Kommando initialisiert die CPU, wodurch sie in einen definierten Anfangszustand gebracht wird.

Beispiel:

```
>>>I

        INIT SEQ DONE
>>>
```

LINK (LI)

Syntax: LINK<CR>

Eröffnen und Beschreiben eines Indirekt-Kommado-Files. Das Konsolprogramm zeigt mit der Ausschrift '<<<' den LINK-Modus an. Alle nachfolgend eingegebenen Kommandos werden in dem Indirekt-Kommando-File abgespeichert. CTRL/C beendet das LINK-Kommando (vgl. PERFORM). Das Indirekt-Kommando-File hat eine normierte Größe von 10 Sektoren und wird stets auf der Konsoldiskette angelegt, unabhängig vom Zustand des Floppy-Drivers.

LOAD (L)

Syntax: LOAD[<qualifier list>]<blank><file specification><CR>

Dabei gilt: <qualifier list> gibt den Speicherbereich für die zu ladende Datei an;
<file specification> ist die Identifikation für die zu ladende Datei.

Dieses Kommando wird zum Lesen oder Übertragen von File-Daten von der Konsol-Floppy zum Hauptspeicher oder zum ladbaren Mikroprogrammspeicher (WCS) benutzt. Wird kein Qualifier angegeben, ist der Standardbereich der physische Hauptspeicher ab Adresse 0; dort beginnt das Laden.

LOAD-Qualifier:

Qualifier	Wirkung
/START: <address> (S)	Angabe einer Startadresse für das Laden.
/WCS (WC)	Die spezifizierte Datei wird in den WCS geladen.
/PHYSICAL (P)	Die spezifizierte Datei wird in den Hauptspeicher geladen.

Beispiel:

```
>>>L/WC WCS124.PAT

        LOAD DONE, 0800 MICROWORDS LOADED
        VER: PCS=01 WCS=0E-10 FPLA=0E CON=V01-00-L
>>>L WCS124.PAT

        LOAD DONE, 00009200 BYTES LOADED
>>>L/ST:4000 WCS124.PAT

        LOAD DONE, 00009200 BYTES LOADED
>>>
```

NEXT (N)

Syntax: NEXT[<blank><count>]<CR>

Das NEXT-Kommando bewirkt, daß der CPU-Takt um eine durch <count> angegebene Anzahl von Schritten weiterläuft. Der Schrittmodus wird durch das vorangegangene Kommando SET STEP bestimmt. Wenn ein NEXT-Kommando eingegeben wird, während die CPU im Normaltakt-Modus läuft, führt sie <count> Schritte im Einzelbefehlsschritt-Modus aus (<count> max. 63, d.h. %X3F bzw. %O77; zur Darstellung von Argumenten siehe Abschnitt 3.11.4.)

OVERLAY (O)

Syntax: OVERLAY<blank><filespecification><CR>

Dieses Kommando bewirkt das Laden des durch <filespezifikation> angegebenen Programms von der Konsoldiskette in den Speicher des Konsolrechners K 1620 sowie dessen Start.

PERFORM (P)

Syntax: PERFORM<CR>

Abarbeiten eines mit LINK erstellten Indirekt-Kommando-Files von der Konsoldiskette, unabhängig vom Zustand des Floppy-Drivers.

QUAD CLEAR (QCLEAR)

Syntax: QCLEAR<blank><physical address><CR>

Die Bits 0-2 (3 niederwertigste Bits) der angegebenen physischen Hauptspeicheradresse <physical address> werden auf Null gesetzt. Das Quadwort an der so erhaltenen Adresse wird gelöscht. Dieses Kommando wird zum Löschen eines nichtkorrigierbaren ECC-Fehlers benutzt.

REBOOT (REB)

Syntax: REBOOT<CR>

Das Konsolprogramm wird wie beim Warmstart neu geladen. Alle mit dem SHOW-Kommando angezeigten Variablen werden auf Standardwerte gestellt. Das Kommando kann bei laufendem SVP gegeben werden, und die Konsoldiskette kann von Laufwerk 0 auf Laufwerk 1 und umgekehrt getauscht werden. Der WCS wird nicht neu geladen.

REPEAT (R)

Syntax: REPEAT<blank><console command><CR>

Das angegebene Konsolkommando wird solange wiederholt, bis ein CTRL/C eingegeben wird. Als <console command> kann - außer REPEAT selbst - jedes Konsolkommando eingegeben werden.

Beispiel:

```
>>>R E 4000

        P 00004000 xxxxxxxx
          .
          .
          .
        P 00004000 xxxxxxxx
>>>^C
```

SET CLOCK (SE C)

Syntax: SET<blank>CLOCK[<blank><clock option>]<CR>

Legt den CPU-Takt auf die durch die Clock-Option bestimmte Frequenz fest. Ohne Angabe eines Arguments wird N = 10.0 MHz eingestellt.

Optionen (<clock option>):

Name	Frequenz
SLOW (S)	8.925 MHz
FAST (F)	10.525 MHz
NORMAL (N)	10.0 MHz

Beispiel:

```
>>>SE C S

>>>SHOW

        CPU HALTED, SOMM CLEAR, STEP=NONE, CLOCK=SLOW
          .
          .
          .

>>>SE C

>>>SH

        CPU HALTED, SOMM=CLEAR, STEP=NONE, CLOCK=NORM
          .
          .
          .

>>>
```

SET DEFAULT (SE D)

Syntax: SET<blank>DEFAULT[<blank><default option>]<CR>

Mit diesem Kommando werden die Standardwerte für Adresse, Datenlänge und Zahlenbasis für Terminal-Ein-/Ausgaben gesetzt.
Enthält ein Konsolkommando keinen Qualifier, verwendet das Konsolprogramm die mit diesem Kommando gesetzten Standardwerte.

Optionen (<default option>):

Adressen-Standards:	VIRTUAL (V), PHYSICAL (P), GENERAL (G), INTERNAL (I), IDBUS (ID), VBUS (VB), CONSOLE (CON)
Daten-Standards:	BYTE (B), WORD (W), LONG (L), QUAD (Q)
Zahlenbasis-Standards:	HEX (H), OCTAL (O)

Ohne Angabe einer Option wird PHYSICAL, LONG, HEX eingestellt.

Beispiel:

```
>>>SE D Q

>>>SH

        CPU HALTED, SOMM=CLEAR, STEP=NONE, CLOCK=NORM
        RAD=HEX, ADD=PHYS, DAT=QUAD, ...
          .
          .
          .
>>>
```

SET RELOCATION (SE R)

Syntax: SET<blank>RELOCATION:<data><CR>

Schreibt <data> in das Verschiebungsregister des Konsolprogramms. Der Inhalt des Verschiebungsregisters wird zur effektiven Adresse von allen EXAMINE- und DEPOSIT-Kommandos für den virtuellen und physischen Hauptspeicher addiert.

SET SOMM (SE SO)

Syntax: SET<blank>SOMM<CR>

Dieses Kommando aktiviert den Mikrobefehlsstop durch Setzen des Flags SOMM (STOP ON MICROBREAK MATCH) auf dem Konsolinterface. Der CPU-Takt wird gestoppt, wenn SOMM gesetzt ist und der Inhalt des Mikrobreak-Registers (ID-Register mit Adresse %X21; zur Darstellung von Argumenten siehe Abschnitt 3.11.4.) mit der aktuellen Mikrobefehlsadresse der CPU K 2810 übereinstimmt.

Beispiel:

```
>>>SE SO

>>>SH

        CPU HALTED, SOMM SET, ...
          .
          .
          .
>>>CL SO

>>>SH

        CPU HALTED, SOMM CLEAR, ...
          .
          .
          .
>>>
```

SET STEP (SE S)

Syntax: SET<blank>STEP[<blank><step option>]<CR>

Dieses Kommando setzt den CPU-Taktschritt-Modus.

Optionen (<step option>):

Name	Wirkung
INSTRUCTION (I)	Setzt CPU-Takt in den Einzelbefehlsschritt-Modus
BUS (B)	Setzt CPU-Takt in den Einzel-ISB-Buszyklusschritt-Modus (jeweils ein ISB-Zyklus wird ausgeführt) ISB = Interner Synchronbus
STATE (S)	Setzt CPU-Takt in den Einzel-ISB-Taktphasenschritt-Modus (jeweils eine ISB-Taktphase ausgeführt)

Ohne Angabe einer Option wird jeder eingestellte Taktschritt-Modus gelöscht; die CPU arbeitet im normalen (free running) Modus (identisch mit Konsolkommando CLEAR STEP).

SET TERMINAL FILL (SE T F)

Syntax: SET<blank>TERMINAL<blank>FILL:<count><CR>

Dieses Kommando legt die Anzahl (<count>) der Leerzeichen fest, die nach einem <CR> (carriage return) oder einem <LF> (line feed) zum Konsol-Terminal übertragen werden.

SET TERMINAL PROGRAM (SE T PR)

Syntax: SET<blank>TERMINAL<blank>PROGRAM<CR>

Dieses Kommando versetzt das Konsolsubsystem in den Programm-E/A-Modus.

SHOW (SH)

Syntax: SHOW<CR>

Vom Konsol-Terminal wird folgendes angezeigt:

a. die gesetzten Standardwerte für Datenlänge, Adreßtyp, Zahlenbasis für Datenein- und -ausgaben,
b. die Anzahl der Terminal-Füllzeichen und der Inhalt des Verschiebungsregisters,
c. der CPU-Status einschließlich dem RUN-/HALT-Status und dem aktuellen Taktmodus,
d. die aktuelle Beziehung von physischen und logischen Bezeichnern der Floppy-Laufwerke.

Beispiel:

```
>>>SH

        CPU HALTED, SOMM CLEAR, STEP=NONE, CLOCK=NORM
        RAD=HEX, ADD=PHYS, DAT=LONG, FILL=00, REL=00000000
        DRIVE_0=DIS, DRIVE_1=DX0=(CONS)
        HALTED AT xxxxxxxx
>>>
```

SHOW VERSION (SH V)

Syntax: SHOW VERSION<CR>

Anzeige der Versionen von Konsolprogramm und Mikroprogramm.

Beispiel:

```
>>>SH V

        VER: PCS=01 WCS=0E-10 FPLA=0E CON=V01-00-L
>>>
```

START (S)

Syntax: START[/WCS]<blank><address><CR>

A. Ohne Qualifier /WCS

Das Kommando führt das Äquivalent der folgenden Konsolkommandofolge aus (Starten eines Hauptspeicherprogramms):

1. Initialisierung der CPU,
2. Laden der <address>-Daten in den PC,
3. Ausgabe eines CONTINUE-Kommandos an die CPU.

B. Mit Qualifier /WCS (/WC)

Das Kommando führt das Äquivalent der folgenden Konsolkommandofolge aus (Starten eines Mikroprogramms):

1. Schreiben der <address>-Daten in das Mikrobefehlsadreßregister
2. Start des CPU-Taktes im normalen Operationsmodus.

TEST (T)

Syntax: TEST[/COM][<device-name>]<CR>

Dieses Kommando aktiviert den Mikrodiagnose-Monitor. <device-name>=DX0 (oder ohne Angabe) bzw. DX1 gibt das Konsol-Floppy-Laufwerk an, in dem die Mikrodiagnose-Diskette liegt. Wird <device name> nicht angegeben,wird der Mikrodiagnose-Monitor auf dem Laufwerk gesucht, auf dem die Konsoldiskette liegt. Ist kein /COM-Qualifier angegeben, wird die Mikrodiagnose sofort gestartet. Nach erfolgreicher Beendigung des Mikrodiagnosetests (d.h. es wurden keine Fehler gefunden) wird der Konsolkommando-Modus wieder eingenommen.
Ist der /COM-Qualifier angegeben, nimmt der Mikrodiagnosemonitor seinen Kommandomodus (MIC>) ein und wartet auf ein Nutzerkommando.

TYPE (TY)

Syntax: TYPE <filespezifikation><CR>

Ausgabe eines beliebigen Disketten-Textfiles auf dem Konsolterminal.

UNJAM (U)

Syntax: UNJAM<CR>

Bewirkt ein Rücksetzen des internen Synchronbusses (ISB).

WAIT DONE (WA D)

Syntax: WAIT<blank>DONE<CR>

Das WAIT-Kommando wird von einer indirekten Kommandodatei ausgeführt. Nach Ausführung wird die Abarbeitung dieser Kommandodatei solange unterbrochen, bis eines der folgenden Ereignisse eintritt:

a. Ein DONE-Signal wird von einem in der CPU laufenden Programm empfangen; dann setzt das Konsolprogramm die Abarbeitung der Kommandodatei fort.
b. Das Konsolprogramm gibt <@EXIT> aus und bricht die Ausführung der Kommandodatei ab, wenn die CPU in den HALT-Zustand geht und kein DONE-Signal empfangen wurde.
c. Der Nutzer gibt ein CTRL/C ein. Das Konsolprogramm bricht die Ausführung der Kommandodatei ab.

WCS (W)

Syntax: WCS[<blank><device-name>]<CR>

Aufruf des Mikro-Debuggers:
Mit <device-name>=DX0 (oder ohne Angabe) bzw. DX1 wird das Laufwerk angegeben, in dem die Mikro-Debugger-Diskette liegt.

3.8. Konsolkommandos, die bei laufender CPU ausgeführt werden

Die meisten Konsolkommandos erfordern, daß die CPU vor der Kommandoausführung in den HALT-Zustand gebracht wird. Einige Konsolkommandos jedoch erfordern keine Wechselwirkung mit der CPU und können deshalb auch bei laufender CPU ausgeführt werden. Diese Kommandos sind:

SHOW
HELP
alle SET-Kommandos
EXAMINE/VBUS
HALT
CLEAR
WAIT

Wenn bei laufender CPU irgendein anderes Kommando angewiesen wird, wird es abgewiesen, und die folgende Fehlerausschrift erscheint auf dem Terminal:

```
        ? CPU NOT IN CONSOLE WAIT LOOP, FUNCTION ABORTED
>>>
```

3.9. Kommentare in Kommandos

In jeder Kommandozeile können Kommentare, gekennzeichnet durch ein Ausrufezeichen (!), verwendet werden. Wenn das Konsolsubsystem ein '!' erkennt, wird der gesamte restliche Text der Kommandozeile ignoriert. Ein Kommentar kann an jeder beliebigen Stelle in der Kommandozeile beginnen (also auch ganz am Anfang). Das '!' ist ein Teil des Kommentars.

Beispiel:

```
>>>! THIS IS A VALID COMMENT

>>>
```

(Kommentar direkt hinter der Eingabeanforderung)

```
>>>EXAMINE 1234! THIS IS ALSO A COMMENT

>>>
```

(Kommentar hinter einem Kommando, vor dem <CR> !)

3.10. Steuerzeichen und Spezialzeichen

Es gibt eine Reihe von Steuerzeichen und Spezialzeichen, die im RVS-Rechnersystem K 1840 eine Reaktion zur Folge haben. Die in Tabelle 3-4 aufgeführten Zeichen sind Bestandteil der Konsolkommandosprache.

Tabelle 3-4: Steuerzeichen und Spezialzeichen der Konsolkommandosprache

Zeichen	Wirkung
CTRL/C	Jede sich ständig wiederholende Konsoloperation wird angehalten: a. nach REPEAT-Kommando b. durch /NEXT-Qualifier c. Verzögerungen durch WAIT-Kommando d. aufeinanderfolgende Schritte nach einem NEXT-Kommando e. Die weitere Abarbeitung eines indirekten Kommando-Files wird nach Beendigung des gerade bearbeiteten Befehls abgebrochen. Das Konsolsubsystem reagiert mit einer Eingabeanforderung.
CTRL/O	Konsol-Terminal-Ausgaben werden unterdrückt bzw. (bei erneuter Eingabe von CTRL/O) ermöglicht. Nach der nächsten Eingabeanforderung sind Terminal-Ausgaben immer möglich.

Tabelle 3-4: (Forts.) Steuerzeichen und Spezialzeichen der Konsolkommandosprache

Zeichen	Wirkung
CTRL/U	Eingabe CTRL/U vor einem Zeilenbegrenzer: Alle Zeichen, die nach dem letzten Zeilenbegrenzer eingegeben sind, (d.h. alle Eingaben in der aktuellen Kommandozeile) werden ignoriert. Das Konsolprogramm reagiert mit: ^U<LF>
DELETE	Drücken der DELETE-Taste löscht das letzte Zeichen der Eingabezeile. Nur Zeichen nach dem letzten Zeilenbegrenzer können gelöscht werden. Mehrere Zeichen können hintereinander durch mehrere DELETE gelöscht werden. Durch das erste DELETE wird ein '', gefolgt vom gelöschten Zeichen angezeigt. Darauf folgende DELETE haben nur die Ausgabe des gelöschten Zeichens zur Folge. Das nächste eingegebene Zeichen (nicht DELETE) hat die Ausgabe eines eines weiteren '', gefolgt vom neuen Zeichen, zur Folge.
CTRL/P	Das Konsolsubsystem geht in den Konsolkommando-Modus über (außer, wenn der Konsolmodus-Schalter in der Stellung LOCAL/DISABLE oder REMOTE/DISABLE ist).
CARRIAGE RETURN (CR)	Beenden einer Konsolkommandozeile.

3.11. Kommando-Qualifier

Qualifier werden in der Kommandozeile zur genaueren Definition des Konsolkommandos eingegeben. Der folgende Abschnitt beschreibt die Qualifier für Adreß- und Datenargumente: Adreßtyp, Datenlänge, numerische Zahlenbasis sowie Adreßfolgen. Fehlen im Konsolkommando mögliche Qualifier, substituiert die Software automatisch entsprechende Standardeinstellungen. Die Standardeinstellungen für Adreßtyp, Datenlänge und Zahlenbasis können durch das SET DEFAULT-Kommando erfolgen; <default adress> wird wie folgt berechnet:

<default adress> = <im letzten EXAMINE- oder DEPOSIT-Kommando benutzte Adresse> + n

Für physische und virtuelle Hauptspeicheradressen gilt:

n	letzter Zugriff auf
1	Byte
2	Wort
4	Langwort
8	Quadwort

Ansonsten ist n = 1. Bei Adressen von allgemeinen Registern folgt auf das allgemeine Register R15 bzw. PC das Register R0.

3.11.1. Qualifier für den Adreßtyp

Diese Qualifier geben den Adreßtyp als virtuelle bzw. physische Hauptspeicheradresse, Registeradresse oder Adresse im Speicher des Konsolrechners an:

```
/VIRTUAL (V)
/PHYSICAL (P)
/IDBUS (ID)
/VBUS (VB)
/GENERAL (G)
/INTERNAL (I)
/CONSOLE (CON)
```

Beispiel - Anzeige der physischen Adresse 1234

```
>>>EXAMINE/PHYSICAL 1234          Nutzereingabe am Terminal

        P 00001234 0000ABCD       Antwort des Systems
>>>
```

Einige Register haben eindeutige Mnemonics; deshalb ist der Qualifier '/GENERAL' bei Adressen wie PC und SP überflüssig:

```
>>>E PC                           (statt EXAMINE/GENERAL PC)

        G 0000000F 00000001
>>>
```

Wenn sich ein Konsolkommando auf eine nichtexistente virtuelle Adresse bezieht, wird die Abarbeitung des Kommandos mit einer Fehlermeldung auf dem Konsol-Terminal abgebrochen.

3.11.2. Qualifier für Datenlänge

Diese Qualifier geben die Länge des Datums, auf das im Kommando zugegriffen wird, an:

```
/BYTE (B)
/WORD (W)
/LONG (L)
/QUAD (Q)
```

Beispiel:

```
>>>EXAMINE/BYTE 1234

        P 00001234 CD
>>>
```

Das Datum (CD) im Byte mit der Adresse 1234 wird auf dem Terminal ausgegeben.

3.11.3. Qualifier für Zahlenbasis

Diese Qualifier geben die Zahlenbasis der Terminal-Ein- bzw. Ausgaben an:

/HEX (H)
/OCT (O)

3.11.4. Lokales Überschreiben der Zahlenbasis

Die aktuelle Standard-Zahlenbasis kann durch einen Präfix zum numerischen Argument in jedem Kommando lokal überschrieben werden:

'%O' für oktale Argumente
'%X' für hexadezimale Argumente

Beispiele:

```
>>>EXAMINE %O7654           (bei Standardzahlenbasis H)

        P 00000FAC FFFF
>>>

bzw.

>>>EXAMINE %X7654           (bei Standardzahlenbasis O)

        P 00000073124 177777
>>>
```

Ein lokales Überschreiben der Zahlenbasis wird durch die beiden ersten Stellen des numerischen Argumentes angewiesen. Diese dürfen nicht durch Leerzeichen oder Tabulatoren vom eigentlichen Argument getrennt werden.

3.11.5. NEXT-Qualifier (/N)

Syntax: <consol command>/NEXT[:<count>]

Das <count>-Argument gibt an, wie oft das Kommando nach der ersten Ausführung wiederholt werden soll. Der Standardwert für <count> ist Eins. Als maximaler Wert kann <count>=63, d.h. %X3F bzw. %O77 angegeben werden.

1.Beispiel:

```
>>>EXAMINE/BYTE 12345/NEXT:2

        P 00012345 54
        P 00012346 76
        P 00012347 98
```

wird vom Konsolprogramm wie folgt abgearbeitet:

1. Notwendige Standardwerte besorgen
2. Kommando (mit Standardwerten) abarbeiten. Der Inhalt des Speicherplatzes 12345 wird ausgegeben; die neue <default address> ist 12346
3. Jetzt wird der Qualifier /NEXT ausgewertet. Das Konsolprogramm wiederholt die Operation gemäß dem <count>-Argument. Dabei wird jedes Mal die <default address> verwendet und neu berechnet. In diesem Beispiel ist der letzte Wert der <default address> 12348.

```
>>>EXAMINE/NEXT:2

        P 00012348 01234500
        P 0001234C 00654300
        P 00012350 FFFFFF00
>>>
```

Da hier kein Qualifier für die Datenlänge angegeben ist, wird die Standardeinstellung (Langwort) benutzt, auch für die Neuberechnung von <default address>.

2.Beispiel:

```
>>>EXAMINE/NEXT:5 R13

        G 0000000D F0F0F0F0
        G 0000000E 8C000000
        G 0000000F 00000001
        G 00000000 00000010
        G 00000001 00020000
        G 00000002 08080001
>>>
```

Auf das allgemeine Register R15 (PC) folgt das Register R0.

3.12. Ketten von Kommandos

Das Konsolkommando LINK (vgl. Kommandobeschreibung) versetzt das Konsolprogramm in den LINK-Modus. Damit können die gewünschten zu kettenden Kommandos eingegeben werden.
Das LINK-Kommando darf nur am Beginn einer Kommandofolge (i.allg. in der ersten Kommandozeile) eingegeben werden. Es wird jeweils eins der zu kettenden Kommandos pro Kommandozeile eingegeben, diese wird mit <CR> abgeschlossen. Das Konsolprogramm meldet sich dann mit der LINK-Modus-Eingabeanforderung ('<<<') und fordert das nächste Kommando an. Die LINK-Operation wird durch Eingabe von CTRL/C beendet.
Die gekettete Kommandofolge wird auf der Konsoldiskette gespeichert und als indirektes Kommando-File behandelt.

Wenn die Eingabe zu kettender Kommandos durch CTRL/C abgeschlossen ist, können keine weiteren Kommandos mehr hinzugefügt werden. Die geketteten Kommandos werden durch Eingabe eines PERFORM-Kommandos einmalig oder wiederholt abgearbeitet. Wenn das PERFORM-Kommando nach CTRL/C angewiesen wird, wird die gekettete Folge einmal abgearbeitet. Wird PERFORM jedoch auch als letztes Kommando in der geketteten Liste angewiesen, wird diese Folge bis zu einem weiteren CTRL/C wiederholt.
Wenn eine Kommandokette inkorrekt eingegeben wurde, gibt das Konsolprogramm bei der Abarbeitung des Kommandos, das einen syntaktischen Fehler enthält, eine Fehlermeldung aus. Ein CTRL/U weist diese Kommandozeile ab.

Beispiel:

```
>>>LINK                         LINK-Modus einstellen

<<<INITIALIZE                   INITIALIZE-Kommando

<<<EXAMINE/LONG 00001234        EXAMINE-Kommando

<<<PERFORM                      PERFORM als Teil der
                                 Kommandoliste
<<<^C                           nach CTRL/C - Ende der Liste

>>>PERFORM                      Ausführen der Liste

   E/L 00001234

        P 00001234 12A00123     Ausführen des EXAMINE

   E/L 00001234

        P 00001234 12A00123     Wiederholen wegen PERFORM
                                 in Liste


   ^C                           Abbruch durch CTRL/C
```

3.13. Fehlermeldungen des Konsolsubsystems

Dieser Abschnitt listet alle Fehlermeldungen des KSS auf und definiert ihr Format und ihre Bedeutung. Um sie von anderen Mitteilungen zu unterscheiden, wird allen Konsolfehlermeldungen ein Fragezeichen vorangestellt. Wo ein Nutzereingriff gefordert wird, erscheinen die notwendigen Schritte in Klammern nach der entsprechenden Fehlerbeschreibung.

3.13.1. Syntaktische Fehler

?'<TEXT-STRING>' IS INCOMPLETE — <TEXT-STRING> ist kein vollständiges Konsolkommando.

?'<TEXT-STRING>' IS INCORRECT — <TEXT-STRING> wird nicht als gültiges Kommando erkannt.

?FILE NAME ERR	Ein <FILENAME> im Kommando kann nicht in RAD50-Code übersetzt werden. (<FILENAME> ist ungültig.)
?IND-COM ERR	Das Konsolprogramm erkennt einen Fehler im Format einer indirekten Kommandodatei. Mögliche Fehler sind: 1. mehr als 80 Zeichen in einer indirekten Kommandozeile oder 2. eine indirekte Kommandozeile wurde nicht mit einem CARRIAGE RETURN (CR) abgeschlossen.

3.13.2. Kommando-Fehler

?FILE NOT FOUND	Der in einem angegebene <filename> stimmt mit keiner Datei auf der mit <device-name> angesprochenen Diskette überein. Dieser Fehler kann auch bei 'HELP', entsprechenden Dateien auf der Diskette nicht gefunden werden.
?NO CPU RESPONSE	Im Konsolprogramm tritt ein TIME OUT beim Warten auf eine Antwort der CPU K 2810 auf. (Bei Wiederholung werden mögliche Hardware-Fehler angezeigt.)
?CPU NOT IN CONSOLE WAITLOOP, COMMAND ABORTED	Ein Konsolkommando, das eine Reaktion der CPU K 2810 erfordert, wurde angewiesen, während die CPU K 2810 nicht in der Konsolservice-Schleife war. (HALT CPU; Wiederholung des Kommandos.)
?CPU CLOCK STOPPED, COMMAND ABORTED	Ein Konsolkommando, das den Takt der CPU K 2810 im Normalmodus erfordert, wurde bei Schrittbetrieb angewiesen. (Löschen des Schrittbetriebes; Wiederholung des Kommandos.)
?CANT DISABLE BOTH FLOPPY's, FUNCTION ABORTED	Es wurde versucht, sowohl den lokalen als auch den fernaufgestellten Folienspeicher inaktiv zu machen.

3.13.3. Mikroroutinen-Fehler

Das Konsolsubsystem benutzt verschiedene Mikroprogramm-Routinen im Mikroprogrammspeicher der CPU K 2810, um Konsolfunktionen auszuführen. Die folgenden Fehler werden durch Mikroroutinenstörungen erzeugt:

?MIC-ERR ON FUNCTION	Mikrobefehlsfehler, der während der Bedienung einer Konsolforderung auftrat; ISB-Fehlerregister werden angezeigt, nachdem diese Mitteilung ausgegeben wurde (Aktion ist vom Fehler abhängig).
?INT-REG ERR	Mikrobefehlsfehler, der beim Versuch, auf ein internes Register der CPU K 2810 zuzugreifen, auftrat; eine illegale Adresse kann diesen Fehler verursachen.

?MICRO-ERROR, CODE=X	Ein unbekannter Mikrobefehlsfehler trat auf. Der von der CPU K 2810 gemeldete Code ist nicht in der Liste der bekannten Fehlercodes.'X' ist der von der CPU K 2810 abgegebene Code.
?MEM-MAN FAULT, CODE=XX	Ein EXAMINE oder DEPOSIT mit virtueller Adresse verursachte einen Fehler in der Speicherverwaltungs-Mikroroutine. erzeugt wird, mit folgenden Bit-Bedeutungen (Bits von rechts numeriert): Bit 0 = Längenverletzung Bit 1 = Fehler trat bei einer PTE-Referenz auf (PTE = Page-Tabellen-Eintrag) Bit 2 = Write- oder Modify-Versuch Bit 3 = Zugriffsverletzung Bit 4...7 = ignorieren

3.13.4. Fehler der CPU K 2810

?INT-STACK INVALID	Die CPU K 2810 ging in den Halt-Zustand, weil der Interrupt-Stack als ungültig markiert war.
?CPU DOUBLE-ERR HALT	Ein Maschinenfehler trat auf, bevor ein vorausgegangener behandelt worden war; er veranlaßt die CPU K 2810, ein "DOUBLE ERROR"-Halt auszuführen. (Es sind die ID-Register %X30 -%X3F zu prüfen; die Inhalte werden zur Fehlerortung herangezogen.)
?ILL I/E VECTOR	Die CPU K 2810 erkannte einen illegalen Interrupt/Exception-Vektor.
?NO USR WCS	Die CPU K 2810 erkannte einen Interrupt/Exception-Vektor zum Nutzer-Mikroprogrammspeicher (WCS), aber es existiert keiner.
?NO USR WCS	Die CPU K 2810 erkannte einen Interrupt/Exception-Vektor.
?CHM ERR	Ein Befehl zur Modusänderung wurde im Zusammenhang mit dem Interrupt-Stack versucht.
INT PENDING	Dies ist eigentlich kein Fehler, aber es wird angezeigt, daß eine Fehlerbedingung anliegt, während vom Konsolprogramm ein HALT-Kommando gegeben wurde. (Fortsetzen der Arbeit der CPU K 2810, um Interrupt zu löschen.)
?MICRO-MACHINE TIME OUT	Zeigt an, daß die K 1840-Mikroprogrammsteuerung Interrupts nicht innerhalb der vorgegebenen maximalen Zeitspanne abfragen konnte.

3.13.5. Folienspeicher-Fehler

DXi NOT ENABLED	Der logische Name 'DXi' ist keinem Floppy-Laufwerk zugewiesen (Mit 'SHOW' prüfen, ggf. 'ENABLE DUAL eingeben).
?FLOPPY NOT READY	Beide Laufwerke nicht ansprechbar. (Fehler der AFS, Ausfall der Stromversorgung der Laufwerke.)
?FLOPPY NOT READY DRIVE_x	a) Beim Bootstrap des Konsolprogramms: Drive x nicht installiert oder offen oder keine Diskette eingelegt. Im Fall x = 0 wird automatisch zu x = 1 übergegangen, im Fall x = 1 wird auf die Eingabe irgendeines Zeichens gewartet, nachdem das gewünschte LW bereitgemacht wurde. Die Prüfung beginnt wieder mit x = 0. b) Bei laufendem Konsolprogramm bzw. SVP: Präzisierungen s. u. y = 1.
?NO BOOT ON FLOPPY DRIVE_x	Von LW x wurde ein Boot-Prozeß versucht. Die dort liegende Diskette enthält aber keinen gültigen Boot-Block; bei x = 0 erfolgt automatischer Übergang zu x = 1; bei x = 1 wird nach Diskettenwechsel auf die Eingabe irgendeines Zeichens über Tastatur gewartet.
?FLOPPY ERROR ON BOOT	Hardwarefehler beim Lesen des Boot-Sektors durch den Anfangslader. Wiederholung nach Eingabe irgendeines Zeichens über Tastatur.
?B-NO CONSOL.SYS	Die Directory der zum Bootstrap des Konsolprogramms eingelegten Diskette enthält kein File mit dem Namen CONSOL.SYS. Es erfolgen automatisch unbegrenzt viele Leseversuche.
?B-I/O ERROR	Hardwarefehler beim Bootstrap des CONSOL.SYS (nach Kaltstart, Warmstart oder Kommando REBOOT) während des Lesens der Directory oder des CONSOL.SYS-Files. Es erfolgen automatisch unbegrenzt viele Leseversuche.

?FLOPPY ERROR, CODE=y	Der Konsol-Floppy-Driver erkannte einen Fehler. y = 1: Floppy-Hardware-Fehler (CRC, Parität usw.). War nur die Diskette nicht eingelegt oder das LW nicht geschlossen, erfolgt folgende Präzisierung: a) Im Konsolmodus ?FLOPPY NOT READY DRIVE_x MOUNT FLOPPY, THEN PRESS SPACE BAR Die Eingabe von CTRL/C führt zum Abbruch der Kommandoausführung. Nach 8 erfolglosen Versuchen bricht das Konsolprogramm automatisch ab. b) Beim Arbeiten mit dem Floppy im SVP erscheint lediglich die zusätzliche Information ?FLOPPY NOT READY DRIVE_x auf dem Konsol-Terminal. Die Ausführung des Kommandos wird abgebrochen. An USER-Terminals erscheinen keine zusätzlichen Ausschriften. y = 2: File nicht gefunden, die Ausschrift wird automatisch umgesetzt in 'FILE NOT FOUND' (s. dort). y = 3: Überlauf der Floppy-Driver-Warteschlange. y = 4: Konsolsoftware forderte eine illegale Sektornummer an.

3.13.6. Kompatibilitäts-Fehler

?WARNING-WCS&FPLA VER MISMATCH	Das Mikroprogramm im WCS ist nicht kompatibel mit dem programmierbaren Logikfeld FPLA. Diese Mitteilung wird bei jedem Makroprogramm-START oder -CONTINUE ausgegeben, es erfolgt aber keine weitere Konsolreaktion.
?FATAL-WCS&PCS	Das Mikroprogramm im PCS ist mit demjenigen im WCS nicht kompatibel. Makroprogramm-Start und -CONTINUE werden durch das Konsolsubsystem verhindert.
?REMOTE ACCESS NOT SUPPORTED	Wird ausgegeben, wenn der Konsolmodusschalter die "REMOTE"-Stellung einnimmt und das Konsolsubsystem keine entsprechende Software-Unterstützung erhält.

3.13.7. Konsolprogramm-Fehler

?TRAP-4, RESTARTING CONSOLE	Das Konsolprogramm ging in Timeout-Trap (Zeitüberschreitung). Es wird ein Restart veranlaßt.
?UNEXPECTED TRAP MOUNT CONSOLE FLOPPY, THEN TYPE ^C	Das Konsolprogramm blieb an einem Trap mit ungültigem Vektor hängen. Das Bootstrap des Konsolprogramms wird wiederholt, wenn CTRL/C eingegeben wird.
?Q-BLKD	Die Ausgabewarteschlange im Konsolprogramm ist blockiert. Erneutes Bootstrap des Konsolprogramms wird eingeleitet.

4. Diagnose

4.1. Diagnosesystemstruktur des RVS K 1840

4.1.1. Ebenen des Diagnosesystems

Ebene 1: Auf dem Betriebssystem SVP 1800 basierende Hilfsmittel zur Prüfung des Gesamtsystems von Hard- und Software (Nutzung virtueller QIO).
Dazu gehören die dem SVP zugeordneten Programmpakete
- UETP (User Environment Test Package)
- SDA (System Dump Analyzer)

Ebene 2R: Auf Supervisor aufbauende Diagnoseprogramme, die nur unter SVP arbeiten (Nutzung physischer QIO). Dies sind Diagnoseprogramme, die nicht vom Supervisor im Stand-alone-Modus unterstützt werden und der System-Exerciser. Der Supervisor dient der Steuerung der Diagnoseprogramme der Ebenen 2R, 2 und 3 und liefert den Kontext für die Abarbeitung der Diagnoseprogramme dieser Ebenen.

Ebene 2: Auf Supervisor aufbauende Diagnoseprogramme, die sowohl selbständig als auch unter SVP arbeiten können (Nutzung physischer QIO).
Dazu gehören:
- Bus-interaction-Programm
- Befehlstests
- Formatierungsprogramme für Plattenspeicher
- Programme für Zuverlässigkeitstests

Ebene 3: Auf Supervisor aufbauende Diagnoseprogramme, die nur selbständig arbeiten können.
Dazu gehören:
- Diagnoseprogramme für die funktionelle Prüfung von peripheren Geräten
- Diagnoseprogramme für Reparaturen an peripheren Geräten
- Befehlstests
- Busadaptertests usw.

Ebene 4: Selbständig arbeitende Makrodiagnoseprogramme ohne Supervisorsteuerung. Dies ist z.B. der Grundbefehlstest. Er testet die Grundfunktionen der CPU, die für die Abarbeitung des Supervisors erforderlich sind.

Ebene 5: (Konsolebene) Auf dem Konsolprogramm aufbauende Diagnoseprogramme, die nur selbständig arbeiten können.
Dazu gehören:
- Mikrodiagnoseprogramme
- Konsolprogramm
- ROM-residente Tests, die nach Netzzuschaltung abgearbeitet werden.
- Diagnoseprogramme zum Test des Konsolrechners und seiner peripheren Geräte

4.1.2. Laden der Diagnoseprogramme und deren Steuerung (Übersicht)

Das Diagnosesystem für RVS K 1840 hat eine große Flexibilität bezüglich Laden und Steuerung der verschiedenen Programmebenen.
Beispielsweise können die Programme der Ebenen 2 und 3 sowohl vom Konsol-Folienspeicher als auch von der Systemplatte geladen werden. Die Mikrodiagnosen werden gewöhnlich nur von der Konsoldiskette des KSS geladen.

4.1.3. Zuordnung der Disketten zu den Diagnoseebenen

4.1.3.1. Konsolebene (Diagnoseebene 5)

DSO: Konsolsubsystemtests

Die Diskette DS0 enthält ein für das Konsolsubsystem (KSS), bestehend aus Konsolrechner K 1620, Konsolinterface, 2 Folienspeicher 5.25", Konsol-Terminal K 8911.80 und Systembedienfeld, selbständiges Diagnosesystem.
Den Kern des Diagnosesystems bildet eine Modifikation des Leitprogrammes LPROES der Rechnerfamilie K 1600, genannt LPROEM.

DS1: K 1840 Standardkonsole

- Die Diskette DS1 ist für jeden Systemanlauf erforderlich
- Konsol-Boot für Konsolprogramm und Laden des WCS
- Bootstrap für SVP 1800 von verschiedenen Datenträgern
- Bootstrap für Diagnose-Supervisors von verschiedenen Datenträgern
- Ermöglichen des Restarts des SVP 1800 aus dem Konsol-E/A-Modus
- Start der CRASH-Kommandodatei vom Konsol-E/A-Modus nach Systemabsturz

DS2: K 1840 MIKRODIA #1

DS3: K 1840 MIKRODIA #2

DS4: reserviert für Ergänzungen

Diese Disketten enthalten die Mikrodiagnoseprogramme und deren Monitore.
Das Kapitel "Mikrodiagnose und Tests mittels WCS" des Diagnosehandbuches Rechner beschreibt ausführlich die Monitorkommandos für die Abarbeitung der Mikrodiagnosen.

4.1.3.2. Diagnoseebene 4

Die Diagnoseprogramme PVKAA.EXE und PVKAS.EXE sind die einzigen Programme der Diagnoseebene 4. Sie sind in MACRO 18 geschrieben und arbeiten selbständig ohne Steuerung des Diagnose-Supervisors. Die erfolgreiche Abarbeitung von PVKAA ist die Voraussetzung für den Start des Supervisors.
PVKAA.EXE und PVKAS.EXE befinden sich gemeinsam mit den Architekturbefehlstests

PVKAB.EXE auf der Diskette DS 5.
Eine eingehende Beschreibung von PVKAA.EXE erfolgt im Kapitel "PVKAA Grundbefehlstest" des "Diagnosehandbuches Rechner".
PVKAS ist der Grundspeichertest. Eine eingehende Beschreibung erfolgt im Kapitel "PVKAS Grundspeichertest" des "Diagnosehandbuches Rechner".

4.1.3.3. Diagnoseebenen 3 und 2

DS5:	K 1840 BEFEH #1	Grundspeichertest, Befehlstests
DS6:	K 1840 BEFEH #2	Befehlstests
DS7:	K 1840 BUS + AUTOKONFIG DIAG	Busadaptertest und Autokonfiguration
DS8:	K 1840 DIAG SUPER	Diagnose-Supervisor
DS9:	K 1840 DISK DIAG #1)	
DS10:	K 1840 DISK DIAG #2)	Plattenspeichertests
DS11:	K 1840 DISK DIAG #3)	
DS12:	AMF18 DIAG	AMF18 Tests
DS13:	AR16W DIAG	AR16W Tests
DS14:	RONAS DIAG	RONAS Tests
DS15:	K 1840 TAPE DIAG	Magnetbandtests

Zu den Dateien der Diagnoseebenen 3 und 2 gehören der Diagnose-Supervisor PSSAA.EXE als zentraler Steuermonitor für alle Diagnoseprogramme (Typ.EXE) der Diagnoseebenen 3, 2 und 2R, die Help-Datei des Supervisors (PSSAA.HLP), die Help-Dateien der Diagnoseprogramme und Kommandodateien für den Supervisor. Diese Dateien sind Bestandteil der Diagnose-Directory SYSMAINT auf Diskette für selbständige Offline-Tests abgelegt.
Der Supervisor ist vom Konsol-E/A-Modus ausgehend startbar durch

- o Laden und Start von Diskette
- o Bootstrap von Diskette
- o Bootstrap vom Systemgerät (Plattenspeicher)

Die Disketten ab DS5 sind für die Programme der Diagnoseebenen 3 und 2 vorgesehen (außer PVKAA.EXE und PVKAS.EXE). Die Diskette DS8 als Träger des Diagnose-Supervisors nimmt dabei eine zentrale Stellung ein.
Für die Diagnoseebene 2R brauchen die Programme NICHT auf der Diskette gespeichert sein, da diese Programme einschließlich Supervisor von der Directory SYSMAINT geladen werden. Die Programme der Diagnoseebene 1 sind Bestandteil des Betriebssystems SVP 1800.
Eine genaue Beschreibung der Programme erfolgt in den entsprechenden Kapiteln des "Diagnosehandbuchs Rechner RVS K 1840" bzw. im "Diagnosehandbuch Peripherie RVS K 1840".

4.2. Diagnose-Supervisor

4.2.1. Programm- und Testablaufsteuerkommandos

Diese Kommandos erlauben es dem Operator, Programme und Programmabschnitte auszuwählen und den Testablauf zu steuern.

In der Kommandobeschreibung haben folgende spezielle Zeichen folgende Bedeutung:

< > schließen symbolische Argumente (numerische Ausdrücke oder Zeichenketten) ein

[ ] schließen optionale Argumente ein

! ODER-Funktion innerhalb des Kommandos

/ kennzeichnet einen Qualifier

Minuszeichen am Ende einer Zeile dient als Kennzeichen, daß das Kommando auf der nächsten Zeile fortgesetzt wird.

SET LOAD-Kommando

SET LOAD [<device>:][<directory>]

Das SET LOAD-Kommando legt das Speichergerät und/oder die Directory fest, von dem der Supervisor die Diagnoseprogramme lädt und in der die Help-Dateien stehen. Als Standardladegerät wird das Gerät eingetragen, von dem der Supervisor geladen und gestartet wurde. Im User-Modus (unter SVP) wird das Standardladegerät und die Directory eingetragen, die äquivalent dem Namen SYS$SYSROOT plus dem Ausdruck [SYSMAINT] sind. Im Normalfall ist dies die Systemplatte plus [SYS0.SYSMAINT], d.h. z.B. DBA0:[SYS0.SYSMAINT].
Dieses Kommando wird somit nur bei Änderung des Speichergerätes und im Zusammenhang mit dem LOAD- oder dem RUN-Kommando verwendet.

SHOW LOAD-Kommando

SHOW LOAD

Der Supervisor zeigt den Namen des Speichergerätes und der Directory an, von denen die Diagnoseprogramme mit Hilfe des LOAD-Kommandos geladen werden.

DIRECTORY-Kommando

DIRECTORY [<file-spec>] [<qualifiers>]

Dieses Kommando zeigt den Inhalt der spezifizierten Directory an. Wurde diese nicht spezifiziert, wird die Standard-Directory (siehe SET LOAD-Kommando) aufgelistet. Der Dateiname und/oder -typ kann durch die Wildcard-Zeichen "*" oder "%" ersetzt werden. Werden diese nicht spezifiziert, wird automatisch "*" eingesetzt. Das Gerät muß vorher angeschlossen worden sein (siehe ATTACH).

/WIDE

Der Inhalt wird untereinander ausgegeben. Dies ist erforderlich, wenn die Dateinamen aus vielen Zeichen (max. 252 einschließlich Sonderzeichen) bestehen.

LOAD-Kommando

LOAD <file-spec>

Dieses Kommando lädt die spezifizierte Datei (d.h. ein Diagnoseprogramm) in den Hauptspeicher. Wurden Gerät und Directory nicht angegeben, werden die Standardwerte genommen bzw. die, die durch das SET LOAD-Kommando festgelegt wurden. Wird der Dateityp nicht angegeben, wird der Standardtyp .EXE verwendet. Wird die Version nicht spezifiziert, wird die Datei mit der höchsten Versionsnummer geladen.
Es steht immer nur ein Diagnoseprogramm im Speicher. Mit dem Laden eines Programmes wird das vorher geladene Programm überschrieben.

SHOW TESTS-Kommando

SHOW TESTS

Es werden die Nummern und die Namen der einzelnen Tests des geladenen Diagnoseprogramms ausgegeben.

SHOW SECTIONS-Kommando

SHOW SECTIONS

Es werden die Namen der Sektionen des geladenen Diagnoseprogramms angezeigt. Ist kein Programm geladen, erfolgt eine Fehlerausschrift.

ATTACH-Kommando

ATTACH <UUT-type> <link-name> <generic-device-name>

Bevor ein Diagnoseprogramm gestartet wird, muß der Operator mit Hilfe verschiedener ATTACH-Kommandos die zum Test genutzten Hardware-Einheiten (unit under test - UUT) anschließen und damit dem Supervisor bekanntmachen. Jede UUT wird einmalig angeschlossen durch ihre physische Bezeichnung und durch den Anschluß zur übergeordneten Funktionsgruppe. Die übergeordnete Funktionsgruppe muß vorher bereits angeschlossen worden sein (mit Ausnahme von HUB).

1. Parameter <UUT-Typ>:
   Physische Bezeichnung des Gerätes bzw. der Funktionsgruppe

2. Parameter <link-name>:
   Allgemeiner Gerätename des Gerätes bzw. der Funktionsgruppe, an die die UUT angeschlossen ist.

3. Parameter <generic-device-name>:
   Allgemeiner Gerätename des Gerätes bzw. der Funktionsgruppe, die angeschlossen wird. Der Parameter hat die allgemeine Form ggan: als Gerätename.

gg	zwei Buchstaben des allgemeinen Gerätenamens
a	ein Buchstabe, der den Gerätekontroller spezifiziert
n	eine Ziffer im Bereich 0...255, die ein Gerät aus einer Summe mehrerer gleicher Geräte an der gleichen Link-Baugruppe spezifiziert.

Für einige Hardwarekomponenten sind zusätzlich Informationen erforderlich (z.B. TR-Nummer und BR-Ebene für den RHA40, Kontroller-Nummer für den BM16, CSR-Vektor und BR-Ebene für ein SKRBUS-Gerät). Werden die zusätzlich erforderlichen Informationen nicht eingegeben, erfolgt automatisch eine Eingabeanforderung vom Supervisor.
Die erforderlichen Eingaben sind den jeweiligen Diagnoseprogrammbeschreibungen zu entnehmen. Das ATTACH-Kommando kann mit CTRL/C abgebrochen werden.

DEATTACH-Kommando

DEATTACH <generic-device-name> /ADAPTER=<link-name>

oder

DEATTACH /ADAPTER=<link-name> <generic-device-name>

Das DEATTACH-Kommando bewirkt das Gegenteil des ATTACH-Kommandos, d.h. das spezifizierte Gerät wird abgetrennt. Zu beachten ist, daß beim Abtrennen eines übergeordnete Gerätes alle daran angeschlossenen Geräte mit abgetrennt werden. Es wird eine Mitteilung ausgegeben, welche Geräte abgetrennt wurden.

SELECT-Kommando

SELECT [<qualifiers>] <generic-device-name>[:]-
[,<generic-device-name>[:]...] !ALL

Das zu testende Gerät ist nach dem ATTACH-Kommando mit dem SELECT-Kommando und dem allgemeinen Gerätenamen auszuwählen. Werden mehrere ausgewählt, werden sie in der Reihenfolge der Auswahl nacheinander in den Test einbezogen.

/ADAPTER=<name>

Dieser Qualifier schränkt das auszuwählende Gerätespektrum auf die Geräte ein, die an das Gerät, spezifiziert mit <name>, angeschlossen sind.

/NOALLOCATE

Ist dieser Qualifier angegeben, wird das ausgewählte Gerät nicht dem Diagnosesystem zugeteilt. Dies wird genutzt, wenn das auszuwählende Gerät bereits einem anderen Nutzer zugeteilt wurde. Nach Freigabe des Gerätes durch den anderen Nutzer kann dieses Gerät sofort in den Test einbezogen werden.

DESELECT-Kommando

DESELECT [<qualifiers>] <generic-device-name>[:]-
[,<generic-device-name>[:]...] !ALL

Das spezifizierte Gerät wird aus der Liste der zu testenden Geräte gestrichen. Dazu muß es vorher ausgewählt worden sein.

/ADAPTER=<name>

Diese Qualifier schränkt das zu inaktivierende Gerätespektrum auf die Geräte ein, die an das Gerät, spezifiziert mit <name>, angeschlossen sind.

SHOW DEVICE-Kommando

```
SHOW DEVICE [<qualifiers>] <generic-device-name>[:]-
            [,<generic-device-name>[:]...]
```

Dieses Kommando veranlaßt den Supervisor, die charakteristischen Parameter der spezifizierten Geräte auf dem Konsol-Terminal auszugeben. Sind keine Geräte spezifiziert, werden die Parameter aller angeschlossenen Geräte (mit ATTACH-Kommando) ausgegeben.

/BRIEF

Mit diesem Qualifier wird veranlaßt, daß nur die allgemeinen Gerätenamen und die Hardware-Bezeichnungen ausgegeben werden.

/ADAPTER=<name>

Dieser Qualifier schränkt das auszugebende Spektrum von Geräten auf die ein, die an das Gerät, spezifiziert mit <name>, angeschlossen sind.

SHOW SELECTED-Kommando

```
SHOW SELECTED [<qualifiers>]
```

Es werden die charakteristischen Parameter der Geräte auf dem Konsol-Terminal ausgegeben, die mit dem SELECT-Kommando ausgewählt wurden. Das Ausgabeformat entspricht dem des SHOW DEVICE-Kommandos.

/BRIEF

Mit diesem Qualifier wird veranlaßt, daß nur die allgemeinen Gerätenamen und die Hardware-Bezeichnungen ausgegeben werden.

START-Kommando

```
START [/SECTION:<section-name>][/TEST:<first>[:<last>!/SUBTEST:<num>]]-
      [/PASSES:<count>][<qualifiers>]
```

Das START-Kommando veranlaßt den Supervisor, die Steuerung an die angewählten Tests des Diagnoseprogramms im Hauptspeicher zu übergeben, um die Programmabarbeitung zu beginnen. Dies setzt voraus, daß mit dem LOAD-Kommando ein Diagnoseprogramm vorher geladen wurde.
Jedes Diagnoseprogramm besteht aus diskreten Tests. Diese Tests sind in Sektionen zusammengefaßt in Abhängigkeit von ihrer Funktion, Abarbeitungszeit und ob Bedienerhandlungen erforderlich sind.
Wird im START-Kommando keine Sektion angegeben, werden die Tests der Standardsektion abgearbeitet. Tests der Standardsektion erfordern keine Bedienerhandlungen. Wird im START-Kommando eine Sektion angegeben, werden die Tests abgearbeitet, die dieser Sektion zugeordnet sind.
Die Eingabe TEST wird in zwei verschiedenen Formen genutzt. Wenn die Argumente "first" und "last" spezifiziert sind, werden nacheinander die Tests, beginnend mit "first" und endend mit einschließlich "last", abgearbeitet. Wird das Argument "first" in Verbindung mit SUBTEST angegeben, wird der Test, spezifiziert mit "first", vom Beginn an gestartet und es werden soviel Subtests dieses Tests abgearbeitet, wie unter SUBTEST spezifiziert. Wird an diese Kommandoform die

Eingabe PASSES angeschlossen, so wird der letzte Subtest sovielmal durchlaufen, wie durch PASSES spezifiziert wurde.
Wenn kein Test spezifiziert wurde, werden alle Tests der ausgesuchten Sektion (mit /SECTION: spezifiziert oder die Standard-Sektion) abgearbeitet. Ist in TEST nur ein Argument spezifiziert, werden alle Tests, beginnend ab dem spezifizierten bis einschließlich dem mit der höchsten Nummer, der ausgesuchten Sektion abgearbeitet.
In die Abarbeitung werden nur die Tests einbezogen, die Bestandteil der ausgesuchten Sektion sind. Mit der Eingabe PASSES kann die Anzahl der Durchläufe spezifiziert werden. Ohne Spezifizierung erfolgt ein einmaliger Durchlauf (Standardwert ist 1). Der Maximalwert bei PASSES ist 0 und bedeutet unendlich viele Durchläufe. Alle Zahlen sind dezimal einzugeben.

/QA

Der Supervisor bietet die Möglichkeit, mit Hilfe dieses Qualifier sogenannte Qualitätssicherungsprüfungen auszuführen. Wenn dieser Qualifier gesetzt ist, durchläuft das Diagnoseprogramm verschiedene Schleifen und Tests. Das Programm wird abgearbeitet, bis irgendein QA-Fehler festgestellt wurde oder bis die QA-Prüfung abgearbeitet wurde. Ziel ist eine Automatisierung des Testablaufs.
Die Qualifiers /PASSES, /SECTION und /SUBTEST werden ignoriert. Es werden nur Tests der Standard-Sektion abgearbeitet.

RUN-Kommando

RUN <file-spec>[/SECTION:<section-name>][/TEST:<first>[:<last>!/SUBTEST:<num>]]-
[/PASSES:<count>][<qualifiers>]

RUN ist das Äquivalent zur LOAD- und START-Kommandofolge. Die Bedeutung der Eingabe ist mit der des START-Kommandos identisch.

/QA

Dieser Qualifier hat die gleiche Funktion wie beim START- Kommando.

CONTROL-C-Kommando

CTRL/C

Normalerweise wird die Steuerung vom Diagnoseprogramm zum Kommandozeilen-Interpreter des Supervisors zurückgegeben. Der Supervisor befindet sich dann in einem Kommandowartezustand und meldet sich mit DS> auf dem Konsol-Terminal. Der Operator kann jetzt irgendein zugelassenes Kommando eingeben. CTRL/C ist das einzige Supervisorkommando, das man während der Abarbeitung eines Programmes eingeben kann. Arbeitet das Diagnoseprogramm im Konversationsmodus kehrt die Steuerung nach Eingabe CTRL/C zum Kommando-Interpreter innerhalb des Programms für den Konversationsmodus zurück.

SUMMARY-Kommando

SUMMARY

Mit diesem Kommando wird ein Programm des Supervisors gestartet, das den Statistical-Report ausgibt. Dieses Kommando wird im allgemeinen nach Abarbeitung

des Diagnoseprogramms ausgeführt. Es kann aber sinnvoll sein, den Statistical-Report während der Abarbeitung des Diagnoseprogramms ausgeben zu lassen. Dazu ist CTRL/C einzugeben, um in den Kommandozeilen-Interpreter zurückzukehren. Danach ist das SUMMARY-Kommando einzugeben. Mit dem CONTINUE-Kommando kann die Programmabarbeitung fortgesetzt werden.
Der Statistical-Report beinhaltet den Namen und die Ausgabe des Diagnoseprogramms, die Summe aller durch das Programm (Hard, Soft, System, Device) und durch den Supervisor festgestellten Fehler.

SHOW STATUS-Kommando

SHOW STATUS •

Es werden Name, Ausgabe, Datum, Uhrzeit, Sektion, Fehlerzählzelle, Test, Subtest, Durchlaufzähler und PC-Inhalt des mit CTRL/C unterbrochenen Programms ausgegeben. Wird kein Programm abgearbeitet, erfolgt eine Fehlerausschrift.

CONTINUE-Kommando

CONTINUE

Dieses Kommando veranlaßt die Programmfortsetzung an dem Punkt, an dem es unterbrochen wurde. Das Kommando wird genutzt nach Unterbrechungspunkt, Fehlerhalt oder nach den Kommandos SUMMARY, CTRL/C.

ABORT-Kommando

ABORT

Dieses Kommando veranlaßt einen Sprung in die Cleanup-Routine mit anschließender Rückkehr zum Supervisor, der in den Kommandowarte-Status eintritt und sich mit DS> meldet. Der Operator kann irgendein Kommando mit Ausnahme CONTINUE eingeben.

EXIT-Kommando

EXIT

Im User-Modus wird die Arbeit im Diagnosesystem beendet, das Betriebssystem meldet sich mit $. Alle ausgewählten Geräte werden automatisch inaktiviert. Im Stand-alone-Modus geht das Programm nach Reinitialisierung des Rechners in Halt. Mit CONTINUE vom Konsolsubsystem kann in den Kommandomodus des Supervisors zurückgesprungen werden.

CONTROL-Z-Kommando

CTRL/Z

Dieses Kommando hat die gleiche Wirkung wie das EXIT-Kommando.

CONTROL-Y-Kommando

CTRL/Y

Die Arbeit des Diagnosesystems wird über Interrupt abgebrochen. Im User-Modus ist die Anwendung nicht sinnvoll, da z.B. die Standard-Directory nicht wieder zurückgestellt wird.

HELP-Kommando

HELP

Der Supervisor gibt einen Überblick über die Möglichkeiten und die Nutzung der Help-Dateien.

HELP <topic>

Der Supervisor gibt entsprechend dem spezifizierten Topic Informationen aus, die er der Help-Datei PVSAA.HLP entnimmt.

HELP <diagnostic name> [<topic> [<subtopic>...]]

Es werden Informationen der Help-Datei des spezifizierten Diagnoseprogrammes und des entsprechenden Topics ausgegeben.

Bemerkung:

Die Help-Dateien müssen in der Standard-Directory oder in der Directory stehen, die mit dem SET LOAD-Kommando festgelegt wurde. Die Eingaben können durch Wildcard-Zeichen ersetzt werden.

MEMORY MANAGEMENT-Kommando

Diese Kommandos sind nicht im User-Modus nutzbar.

SET MM ON
Einschalten Speicherverwaltung,

SET MM OFF
Auschalten Speicherverwaltung,

SHOW MM
Anzeige, ob Speicherverwaltung ein- oder ausgeschaltet ist.

SET MEMORY-Kommando

SET MEMORY <n>

Mit <n> kann die Größe des physischen Speichers (in Seiten) festgelegt werden. Dies kann erforderlich sein, wenn der Supervisor und das Diagnoseprogramm in einem kleineren Speicherbereich arbeiten sollen als dem physisch vorhandenen. Der Wert von <n> muß größer oder gleich Null, aber kleiner als der mögliche physische Speicher sein. Das Kommando arbeitet nicht im User-Modus. Angeschlossene oder ausgewählte Geräte sind nach Ausführung des Kommandos inaktiviert und freigegeben.

SHOW MEMORY-Kommando

SHOW MEMORY [<qualifiers>]

Mit diesem Kommando kann die Speicherplatzaufteilung des Diagnoseprogrammsystems angezeigt werden.

/MAP

Ausgabe Start- und Endadresse des Diagnoseprogramms, des Supervisors und des dynamischen Speichers sowie die Größe des physischen Speichers. Dieser Qualifier kann auch weggelassen werden.

/DATA_STRUCTURE

Zeigt die internen Datenstrukturadressen des Diagnosesystems einschließlich aller Stacks, Page-Tabellen, SCB und anderer Datenstrukturen (z.B. PCB) an. Die Kenntnis der Datenstruktur kann wertvoll für die Fehlersuche sein. Zu beachten ist, daß sich die Speicheraufteilung im Stand-alone-Modus von der im User-Modus stark unterscheidet, so daß diese im User-Modus nicht angezeigt wird.

/BUFFER

Zeigt den möglichen Platz im P0-, P1- und System-Adreßraum für Puffer der Diagnoseprogramme sowohl oberhalb, als auch unterhalb des Diagnosesystems an (nur sinnvoll, wenn Diagnoseprogramm geladen ist).
Im User-Modus wird der Speicher oberhalb des Systems nicht angezeigt.

/ALL

Es werden /MAP, /DATA_STRUCTURE und /BUFFER zusammengefaßt.

SET WIDTH-Kommando

SET WIDTH <n>

Mit diesem Kommando kann die Anzahl der Spalten <n> des Konsol-Terminals eingestellt werden, wenn diese kleiner ist als 132. Ist bei der Ausgabe die Spaltenanzahl erreicht, wird automatisch <CR>, <LF> ausgeführt.

SHOW WIDTH-Kommando

SHOW WIDTH

Es wird die eingestellte Anzahl der Spalten des Konsol-Terminals angezeigt.

SET PAGE-Kommando

SET PAGE <n>

Mit <n> wird die Größe einer Terminalseite spezifiziert. Sind <n> Zeilen ausgegeben, wird die Ausgabe gestoppt, und es wird erst nach Quittierung einer Eingabeanforderung weiter ausgegeben. Damit wird ein Verlust der ausgegebenen Daten vermieden. Mit CTRL/C kann die weitere Ausgabe abgebrochen werden. Für einen 24-zeiligen Bildschirm ist <n> gleich 22 zu setzen, um eine maximale

Zeilenanzahl ohne Verlust ausgeben zu können. Die restlichen zwei Zeilen werden für die Eingabeanforderung benötigt. Der Standardwert ist Null, d.h. es wird keine Seiteneinteilung vorgenommen.

SET ENFORCE-Kommando

SET ENFORCE

Alle Gerätenamen müssen vollständig angegeben werden. Nach dem Start des Supervisors ist dies der Grundzustand.

CLEAR ENFORCE-Kommando

CLEAR ENFORCE

Ein Teil der Gerätenamen kann verkürzt angegeben werden.

SHOW CALLS-Kommando

SHOW CALLS

Es werden Frame, PC, AP und PSW der abgearbeiteten Call-Befehle ausgeschrieben.

4.2.2. Scripting

Die Möglichkeit des Scripting erlaubt dem Operator, vorher definierte Kommandofolgen automatisch abzuarbeiten. Die Supervisor-Kommandos, die normalerweise über das Konsol-Terminal eingegeben werden, werden statt dessen aus einer Textdatei genommen.

SCRIPTING-Kommando

@<file-spec>

Dieses Kommando veranlaßt den Supervisor zur Abarbeitung der Kommandos, die in der spezifizierten Datei enthalten sind. Die Datei wird mit dem Texteditor erarbeitet. Zu Beginn jeder Zeile ist DS> (nur mit Großbuchstaben und mit Leerzeichen nach spitzer Klammer!) zu schreiben. Danach wird die Datei in die gleiche Directory kopiert, auf der der Supervisor steht. Steht diese Datei nicht in der gleichen Directory wie der Supervisor, ist das Gerät und die Directory im Kommando zu spezifizieren oder vorher mit dem SET LOAD-Kommando einzugeben. Sind in der Dateispezifikation Typ und Versionsnummer nicht enthalten, nimmt der Supervisor den Standarddateityp .COM und die höchste Versionsnummer.

4.2.3. Abarbeitungssteuerfunktionen

Die Abarbeitungssteuerfunktionen gestatten, die Eigenschaften der Diagnoseprogramme und des Supervisors zu variieren. Diese Funktionen werden über Kommando-Flags und Ereignis-Flags realisiert. Die Kommando-Flags werden genutzt zur Steuerung von Fehlermitteilungen, Signalen und Stop bzw. zur zyklischen Abarbeitung der Programme.

SET FLAGS-Kommando

SET [FLAGS] <arg-list>

Dieses Kommando führt zum Setzen von Abarbeitungs-Steuer-Flags, die in der Argumentenliste spezifiziert sind. Die Argumentenliste besteht aus einer Kette von Mnemonics der Tabelle 4-1, die durch Kommas getrennt werden.

Tabelle 4-1: Abarbeitungs-Steuer-Flags

Mnemonik	Wirkung
HALT	Halt bei Fehler. Wenn das Programm einen Fehler festgestellt hat und dieses Flag ist gesetzt, tritt der Supervisor nach Ausgabe aller Fehlermitteilungen, die in Beziehung zu diesem Fehler stehen, in den Kommandowarte-Status ein. Das Programm kann fortgesetzt, erneut gestartet oder abgebrochen werden. Dieses Flag hat Vorrang vor dem LOOP-Flag.
BINARY	Ist dieses Flag gesetzt, wird vor jede Ausschrift (einschließlich der Eingabeanforderungen und der Diagnoseprogrammausgaben) ein Byte ausgegeben, das die Ausschrift klassifiziert (z.B. Kommandofehler, Programmstart, Hardwarefehler usw.). Der Inhalt des Bytes ist in den Programmen zu definieren. Der Grundzustand des Flags ist zurückgesetzt und es wird durch die Kommandos SET FLAGS ALL oder CLEAR FLAGS ALL beeinflußt.
LOOP	Schleife bei Fehler. Wenn dieses Flag gesetzt ist, wird ein vorher bestimmter Programmabschnitt eines Tests oder Subtests in einer Schleife abgearbeitet, so daß man den Fehler feststellen kann. Wenn das IE1-Flag gesetzt ist, werden die Fehlermitteilungen unterdrückt. Beachte, daß damit aber keine Mitteilungen des Programms oder des Supervisors verboten werden. Die Programmschleife wird solange abgearbeitet, bis der Operator die Steuerung mit CTRL/C an den Supervisor übergibt. Danach kann das Programm fortgesetzt, das Flag gelöscht und das Programm fortgesetzt oder abgebrochen werden.
BELL	Signal bei Fehler. Bei gesetztem Flag sendet der Supervisor jedesmal ein Signal an den Operator, wenn das Programm einen Fehler feststellt.
IE1	Verbieten von Fehlermitteilungen der Ebene 1. Wenn dieses Flag gesetzt ist, werden alle Fehlermitteilungen unterdrückt mit Ausnahme derjenigen, die durch den Supervisor und das Programm erzeugt werden.

Tabelle 4-1: (Forts.) Abarbeitungs-Steuer-Flags

Mnemonik	Wirkung
IE2	Verbieten von Fehlermitteilungen der Ebene 2. Wenn dieses Flag gesetzt ist, werden die Basis- und die erweiterten Informationen, die den Fehler betreffen, unterdrückt. Nur der Kopf der Fehlermitteilung (die ersten drei Zeilen) werden bei jedem Fehler ausgegeben.
IE3	Verbieten von Fehlermitteilungen der Ebene 3. Wenn dieses Flag gesetzt ist, werden die erweiterten Fehlerinformationen unterdrückt. Der Kopf und die Basisinformationen werden für jeden Fehler ausgegeben.
IES	Verbieten Summary-Report. Wenn dieses Flag gesetzt ist, werden die Statistical-Reports unterdrückt.
QUICK	Schnelltest. Das gesetzte Flag zeigt dem Programm an, daß der Operator einen Schnelltestmodus wünscht. Die Interpretation dieses Flags ist programmabhängig.
VERIFY	Ist das Flag gesetzt, werden die von der Script-Datei gelesenen Kommandos einschließlich der Eingabeanforderungen auf das Konsolterminal ausgegeben. Fehler- und sonstige Mitteilungen sowie Kommandos, die von einer SVP-Script-Datei gelesen werden, sind davon nicht betroffen.
TRACE	Mitteilung über die Ausführung jedes Tests. Das gesetzte Flag veranlaßt den Supervisor, die Ausführung jedes einzelnen Tests innerhalb des Programms zu protokollieren, obwohl der Supervisor die Steuerung an den Test abgibt.
OPERATOR	Operator anwesend. Das gesetzte Flag zeigt dem Supervisor an, daß eine interaktive Arbeit mit dem Operator möglich ist. Wenn das Flag zurückgesetzt ist, leitet der Supervisor geeignete Maßnahmen ein, um die Weiterarbeit des Tests ohne Operator sicherzustellen.
PROMPT	Anzeige eines langen Dialogs. Das gesetzte Flag zeigt dem Supervisor an, daß der Operator die Limits und Standards von allen Werten, die durch das Programm ausgegeben werden, angezeigt haben möchte.
SEARCH	Einschalten Fehlersuchmodus. Wenn dieses Flag gesetzt ist, wird automatisch eine Modifikation des Testlimits vorgenommen. Wurde durch einen Test ein Fehler mitgeteilt, wird das nächste Testlimit auf den Wert des vorhergehenden Tests gesetzt. Mit einem unendlichen Zyklenzähler (ist Standard, wenn das SEARCH-Flag gesetzt ist) werden die Testzyklen eventuell soweit reduziert, daß nur noch die Tests einschließlich des Tests mit der kleinsten Nummer, der keinen Fehler brachte, abgearbeitet werden.
ALL	Alle Flags dieser Liste werden gesetzt mit Ausnahme der Flags SEARCH und BINARY, die nicht verändert werden (weder gesetzt noch zurückgesetzt).

CLEAR FLAGS-Kommando

CLEAR [FLAGS] <arg-list>

Dieses Kommando führt zum Löschen der in der Argumentenliste spezifizierten Flags. Alle anderen Flags werden nicht beeinflußt. Die Argumentenliste besteht aus einer Kette von Mnemonics, die durch Komma getrennt werden. Argumente siehe SET FLAGS-Kommando.

SET FLAGS DEFAULT-Kommandos

SET FLAGS DEFAULT

Dieses Kommando setzt die Flags OPERATOR und PROMPT. Alle anderen Flags werden gelöscht.

SHOW FLAGS-Kommando

SHOW FLAGS

Dieses Kommando zeigt alle Abarbeitungs-Steuer-Flags und deren aktuellen Zustand. Die Flags werden in zwei Listen ausgegeben; eine Liste mit den gesetzten und eine Liste mit den gelöschten Flags.

SET EVENT FLAGS-Kommando

SET EVENT [FLAGS] <arg-list> !ALL

Dieses Kommando führt zum Setzen der Ereignis-Flags, die in der Argumentenliste spezifiziert wurden. Die anderen Ereignis-Flags werden nicht beeinflußt. Die Argumentenliste beinhaltet die Nummern der Flags im Bereich 1 - 23, die durch Kommas getrennt werden. Mit ALL werden alle Flags gesetzt.
Die Ereignis-Flags sind Statusbits, die durch das SVP und den Supervisor aufgestellt werden. Die Diagnoseprogramme können die Ereignis-Flags zur Veränderung von Mitteilungsfunktionen, einschließlich der Kommunikation mit dem Operator, nutzen.

Das SVP spezifiziert die Funktionen der ersten zwei Ereignis-Flags.

Ereignis-Flag 1 erlaubt (wenn gesetzt) oder verbietet (wenn gelöscht) Error-Logging unter SVP, Standard ist gelöscht.
Ereignis-Flag 2 erlaubt (wenn gesetzt) oder verbietet (wenn gelöscht) Wiederholungen im Fehlerfall unter SVP. Standard ist gelöscht.

Die anderen möglichen Ereignis-Flags (3 - 23) sind nicht speziell definiert. Die Diagnoseprogramme können die Flags für die interaktive Arbeit mit dem Operator nutzen, wobei deren Funktion in der Programmdokumentation aufzuführen ist. Zum Beispiel kann man ein Flag nutzen, um besondere Datenmuster zu spezifizieren.

CLEAR EVENT FLAGS-Kommando

CLEAR EVENT [FLAGS] <arg-list> !ALL

Dieses Kommando löscht die in der Argumentenliste spezifizierten Ereignis-Flags. Die anderen Flags werden nicht beeinflußt. Die Argumentenliste enthält die Flag-Nummern im Bereich 1 - 23, getrennt durch Komma. Mit ALL werden alle Flags gelöscht.

SHOW EVENT FLAGS-Kommando

SHOW EVENT [FLAGS]

Dieses Kommando veranlaßt den Supervisor, eine Liste aller gesetzten Ereignis-Flags auszugeben.

SET QA FLAGS-Kommando

SET QAMULTIPLEPASS <n>

Mit <n> wird die Anzahl der Durchläufe im Multiple-Pass-Check festgelegt. Standardwert ist 10, d.h. es werden 10mal nacheinander alle Tests abgearbeitet.

SET QATESTLOOPS <n>

Mit <n> wird festgelegt, wievielmal jeder Test abgearbeitet wird. Standardwert ist 100, d.h., es wird 100mal ein Test abgearbeitet, ehe zum nächsten Test übergegangen wird.

SET QASUBTESTLOOPS <n>

Mit <n> wird festgelegt, wievielmal jeder Subtest abgearbeitet wird. Wenn z.B. ein Diagnoseprogramm zwei Tests, Test 1 mit drei Subtests und Test 2 mit zwei Subtests, enthält und QASUBTESTLOOPS ist auf drei gesetzt, dann werden die Tests in folgender Reihenfolge ausgeführt:

1.1, 1.1, 1.1,
1.1, 1.2, 1.2, 1.2,
1.1, 1.2, 1.3, 1.3, 1.3,
1.1, 1.2, 1.3, 2.1, 2.1, 2.1,
2.1, 2.2, 2.2, 2.2
2.1, 2.2

Der Standardwert ist 100.

SET QADEFAULTS

Es werden den eventuell geänderten Flags ihre Standardwerte zugeordnet. Diese sind:

QAMULTIPLEPASS	10
QATESTLOOPS	100
QASUBTESTLOOPS	100

SHOW QA FLAGS-Kommando

SHOW QAMULTIPLEPASS
SHOW QATESTLOOPS
SHOW QASUBTESTLOOPS

Mit diesen Kommandos können die Einstellungen der Flags abgefragt werden.

SHOW QADEFAULTS

Mit diesem Kommando werden die Standardwerte (nicht die evtl. veränderten) der Flags angezeigt.

4.2.4. Debug- und Dienstprogramm-Kommandos

Diese Gruppe von Kommandos gibt dem Operator die Möglichkeit, Fehler einzukreisen und den Code der Diagnoseprogramme zu ändern. Der Supervisor gestattet das gleichzeitige Setzen von bis zu 15 Unterbrechungspunkten innerhalb des Programms. Der Operator kann auch das Image im Hauptspeicher prüfen und modifizieren.

SET BASE-Kommando

SET BASE <address>

Dieses Kommando lädt die spezifizierte Adresse in ein Software-Register. Diese Adresse wird dann genutzt als Basisadresse für die Adresse, die in den Kommandos SET BREAKPOINT, CLEAR BREAKPOINT und DEPOSIT eingegeben werden. Beide werden addiert. Das SET BASE-Kommando ist nützlich bei Vergleichen mit den Codes in den Diagnoseprogrammlisten. Die Basis sollte die Basisadresse (siehe Program-Memory-Allocation-Map) der Programmsektion sein, auf die Bezug genommen wird. Dann kann der PC-Inhalt der Liste direkt für den zu vergleichenden Speicherplatz in der Programmsektion genutzt werden. Bei der Eingabe der Adresse sind die eingestellten Standard-Qualifiers zu beachten.

Bemerkung:

Virtuelle Adresse = physische Adresse (Normalfall), wenn die Speicherverwaltung ausgeschaltet ist.

SHOW BASE-Kommando

SHOW BASE

Mit diesem Kommando kann der Wert der laufenden Basisadresse angezeigt werden.

SET BREAKPOINT-Kommando

SET BREAKPOINT <address>

Dieses Kommando veranlaßt die Übergabe der Steuerung an den Supervisor, wenn der Befehlszähler die Adresse erreicht hat, die vorher durch dieses Kommando eingegeben wurde. Gleichzeitig können maximal 15 Unterbrechungspunkte im Diagnoseprogramm gesetzt werden. Bei der Eingabe der Adresse sind die eingestellten Standard-Qualifiers zu beachten.

CLEAR BREAKPOINT-Kommando

CLEAR BREAKPOINT <address> !ALL

Dieses Kommando löscht den vorher gesetzten Unterbrechungspunkt im Speicher, der durch <address> spezifiziert ist. Existiert auf dieser spezifizierten Adresse kein Unterbrechungspunkt, wird keine Fehlermeldung ausgegeben. Mit ALL werden alle vorher definierten Unterbrechungspunkte gelöscht. Bei der Eingabe der Adresse sind die eingestellten Standard-Qualifiers zu beachten.

SHOW BREAKPOINT-Kommando

SHOW BREAKPOINT

Dieses Kommando zeigt alle z.Z. gesetzten Unterbrechungspunkte an.

SET DEFAULT-Kommando

SET DEFAULT <arg-list>

Mit diesem Kommando werden die Standard-Datenlänge und/oder der Standard-Zahlenbasis-Qualifier für die EXAMINE-, SET BREAKPOINT-, CLEAR BRAEKPOINT-, SET BASE- und DEPOSIT-Kommandos sowie die Qualifiers für die EXAMINE- und DEPOSIT-Kommandos gesetzt. Werden beide angegeben, sind sie durch Komma zu trennen. Wird nur einer angegeben, wird der andere nicht beeinflußt. Nach dem Start des Supervisors sind HEX und LONG die Standard-Qualifiers.

Standard-Qualifiers können sein:

Datenlänge: Byte, Word, Long
Zahlenbasis: Hexadecimal, Decimal, Octal

SHOW DEFAULT-Kommando

SHOW DEFAULT

Mit diesem Kommando werden die eingestellten Standard-Qualifiers angezeigt.

EXAMINE-Kommando

EXAMINE [<qualifiers>] <address>

Das EXAMINE-Kommando zeigt die Speicheradresse und den Inhalt des Speichers in dem Format an, wie es durch die Qualifiers gefordert wird. Sind keine Qualifiers angegeben, werden die Standard-Qualifiers (siehe SET DEFAULT-Kommando) genommen. Die eingegebene Adresse wird nur durch die Standard-Qualifiers beeinflußt.

Mögliche Qualifiers:

/BYTE	Ausgabe eines Bytes
/WORD	Ausgabe eines Wortes
/LONGWORD	Ausgabe eines Langwortes
/HEXADECIMAL	Anzeige Adresse und Daten hexadezimal
/DECIMAL	Anzeige Adresse und Daten dezimal
/OCTAL	Anzeige Adresse und Daten oktal

/ASCII	Anzeige Daten in ASCII-Zeichen und hexadezimal und Adresse hexadezimal
/NEXT=<n>	Anzeige des Inhalts der spezifizierten und der nachfolgenden <n> Speicherplätze, <n> ist eine Dezimalzahl

Als Option ist die Spezifizierung der Adreßargumente mit %D für dezimal, %O für oktal oder %X für hexadezimal vor der Adresse möglich. Adreßausdrücke können auch sein R0 - R11, AP, FP, SP, PC, PSL. Für AP, FP, SP und PC ist auch die Eingabe R12 - R15 möglich. Inhalte interner Prozeßregister können mit dem Adreß-Argument Pxx angezeigt werden, wobei xx die Registernummer ist (Standard-Qualifier beachten).

DEPOSIT-Kommando

DEPOSIT [<qualifiers>] <address> <data>

Mit diesem Kommando können Daten auf einen Speicherplatz, spezifiziert durch <address>, in der durch den Qualifier festgelegten Datenlänge und den Standard-Zahlenbasis-Qualifier festgelegten Datenart geschrieben werden. Ist kein Qualifier spezifiziert, wird der Standard-Qualifier genommen. Die Spezifizierung der Adresse und der Daten mit %D für dezimal, %O für oktal oder %X für hexadezimal vor dem jeweiligen Parameter ist ebenfalls möglich. Inhalte interner Prozessorregister können mit dem Adreßargument Pxx geändert werden, wobei xx die Registernummer ist (Standard-Qualifiers beachten).

Mögliche Qualifier:

/BYTE	Beschreiben eines Bytes
/WORD	Beschreiben eines Wortes
/LONGWORD	Beschreiben eines Langwortes
/NEXT=<n>	Speichert die Daten auf den spezifizierten und die nachfolgenden <n> Speicherplätze. <n> ist eine Dezimalzahl.

NEXT-Kommando

NEXT [<number-of-instructions>]

Dieses Kommando veranlaßt den Supervisor, einen Makrobefehl auszuführen. Ist die Anzahl (dezimal) spezifiziert, werden soviel Makrobefehle abgearbeitet wie angegeben. Der Supervisor zeigt nach Ausführung jedes Befehls den PC des nächsten Befehls und den Inhalt der nachfolgenden vier Bytes an. Dieses Kommando wird zur schrittweisen Abarbeitung von Programmabschnitten, wo Probleme auftreten, genutzt. Das Kommando NEXT ist erst anzuwenden, nachdem das Programm durch einen Unterbrechungspunkt gestoppt wurde.

Abkürzungsverzeichnis

AP	Argument-Pointer
CIB	Konsolinterface
CPU	Zentrale Verarbeitungseinheit
CR	Beenden einer Zeile
CTRL/C	Steuerkommando
E/A	Ein/Ausgabe
FP	Frame-Pointer
HUB	allgemeiner Gerätename des ISB
ID	interne Daten
IEC	internationaler Standard
ISB	interner Synchronbus
KSS	Konsolsubsystem
LED	Leuchtdiode
LF	Zeichenvorschub
PC	Befehlszähler
PCB	Prozeß-Steuerblock
PSL	Programmstatus-Langwort
PSW	Programmstatus-Wort
QA	Qualitätssicherung
QIO	Ein/Ausgabe-Warteschlange
RAM	beschreibbarer Speicher
ROM	Festwertspeicher
RVS	Rechner mit virtuellem Speicher
SCB	System-Steuerblock
SP	Stack-Pointer
SVP	Operating-System-for-virtual-Program-and-Data-Memory
WCS	ladbarer Mikroprogrammspeicher

Sachwortverzeichnis

Bedien- und Anzeigeelemente 2-1
Bootstrap
-Ausführung 2-5
CPU/Speicher- 2-4
Konsolrechner- 2-4
-Programm 2-4, 3-2
-Prozeß 2-3

CPU-Takt 3-2, 3-18

Diagnosekommando
ABORT 4-9
ATTACH 4-5
CLEAR BREAKPOINT 4-19
CLEAR ENFORCE 4-12
CLEAR EVENT FLAGS 4-16
CLEAR FLAGS 4-15
CONTINUE 4-9
CONTROL-C 4-8
CONTROL-Y 4-10
CONTROL-Z 4-10
DEATTACH 4-6
DEPOSIT 4-20
DESELECT 4-7
DIRECTORY 4-4
EXAMINE 4-19
EXIT 4-10
HELP 4-10
LOAD 4-5
MEMORY MANAGEMENT 4-10
NEXT 4-20
RUN 4-8
SCRIPTING 4-13
SELECT 4-6
SET BASE 4-18
SET BREAKPOINT 4-18
SET DEFAULT 4-19
SET ENFORCE 4-12
SET EVENT FLAGS 4-16
SET FLAGS 4-13
SET FLAGS DEFAULT 4-16
SET LOAD 4-4
SET MEMORY 4-11
SET PAGE 4-12
SET QA FLAGS 4-17
SET WIDTH 4-12
SHOW BASE 4-18
SHOW BREAKPOINT 4-19
SHOW CALLS 4-12
SHOW DEFAULT 4-19
SHOW DEVICE 4-7
SHOW EVENT FLAGS 4-17
SHOW FLAGS 4-16
SHOW LOAD 4-4

SHOW MEMORY 4-11
SHOW QA FLAGS 4-17
SHOW SECTIONS 4-5
SHOW SELECTED 4-7
SHOW STATUS 4-9
SHOW TESTS 4-5
SHOW WIDTH 4-12
START 4-7
SUMMARY 4-9
Diagnose-Supervisor 4-3
Diagnoseebenen 4-1, 4-3

Fehlermeldungen, syntaktische 3-27
Fehlermeldungen, Konsolkommando- 3-28
Fehlermeldungen, Mikroroutinen- 3-28
Fehlermeldungen der CPU 3-29
Fehlermeldungen der Folienspeicher 3-30
Fehlermeldungen wegen Inkompatibilitäten 3-31

Konsolkommando
BOOT (B) 3-7
CLEAR SOMM (CL SO) 3-8
CLEAR STEP (CL S) 3-8
CONTINUE (C) 3-8
DEPOSIT (D) 3-9
DIRECTORY (DIR) 3-10
DISABLE DUAL (DIS DU) 3-9
DUPLICATE 3-10
ENABLE DUAL (EN DU) 3-10
EXAMINE (E) 3-11
EXAMINE IR (E IR) 3-11
FORMAT 3-11
HALT (H) 3-12
HELP (HE) 3-12
HELP ABBREV (HE ABBREV) 3-12
HELP ERROR (HE ERR) 3-13
INDIRECT (@) 3-13
INITIALIZE (I) 3-13
LINK (LI) 3-13
LOAD ((L) 3-14
NEXT (N) 3-14
OVERLAY (O) 3-15
PERFORM (P) 3-15
QUAD CLEAR (QCLEAR) 3-15
REBOOT (REB) 3-15
REPEAT (R) 3-15
SET CLOCK (SE C) 3-16
SET DEFAULT (SE D) 3-17
SET SOMM (SE SO) 3-17
SET RELOCATION (SE R) 3-17
SET STEP (SE S) 3-18
SET TERMINAL FILL (SE T F) 3-18
SET TERMINAL PROGRAM (SE T PR) 3-19
SHOW (SH) 3-19
SHOW VERSION (SH V) 3-19
START (S) 3-20
TEST (T) 3-20
TYPE 3-20

UNJAM (U) 3-20
WAIT DONE (WA D) 3-21
WCS (W) 3-21
Ketten von Kommandos 3-26
Kommentare in Kommandos 3-22
Konsol
-Boot 2-3
-E/A-Modus 3-1
-folienspeicher 2-9
-interface (CIB) 2-1
-kommandosprache 3-4
-programm 2-3, 3-4
-steuerfunktionen 3-3
-subsystem (KSS) 3-1

Lokales Überschreiben der Zahlenbasis 3-25

Memory-Sizer 2-4
Mikro-Debugger 3-21
Mikrobefehlsstop 3-17
Mikrodiagnose 3-20

Primär-Boot-Datei 2-3
Primärlader 2-3
Programm-E/A-Modus 3-1

Qualifier
/WCS 3-20
für Adreßtyp 3-24
für Datenlänge 3-24
für Kommandos 3-23
für LOAD 3-14
für Zahlenbasis 3-25
NEXT- (/N) 3-25
Qualitätssicherungsprüfungen 4-8

Restart-Routine 2-4

Sekundärlader 2-4
Standardwerte 3-17
Steuerzeichen und Spezialzeichen 3-22
Stromversorgungsmoduln (STM) 2-1
Systembedienfeld (SBF) 2-1

Terminal-Füllzeichen 3-18

Verschiebungsregister 3-17

WCS-Datei 2-3

Dok.-Nr. 1.57.550020.6/53 10.11.88

Nutzerhandbuch Rechner

Nach der Drucklegung eingetretene Änderungen:

Unter "Einführung" neu aufnehmen:

Gesundheits-, Arbeits- sowie Brandschutz (GAB)

Für den Rechner, bestehend aus CPU-Schrank, SKRBUS-Erweiterungsschrank und Geräteschrank, liegen die GAB-Nachweise im VEB Kombinat Robotron vor.
Schutzgüte gemäß ASVO ist vorhanden.
Der Rechner darf erst nach der Inbetriebnahme und der Abnahmeprüfung gemäß Installationshandbuch Teil 2, Dok.-Nr.1.57.550014.2/54 genutzt werden. Für das arbeits- und brandschutzgerechte Verhalten ist TGL 30513/03 verbindlich. Der Nutzer hat sich mit dem Inhalt dieser TGL vertraut zu machen, insbesondere über das Verhalten bei Havariesituationen. Darüberhinaus sind folgende Festlegungen einzuhalten:

Die Rechnerschränke sind im Nutzerbetrieb geschlossen und verriegelt zu halten. Sie dürfen nur durch unterwiesenes Fachpersonal geöffnet werden (z.B. bei E/A-Schaltung, Diskettenwechsel, Wartung, Fehlersuche, Reparatur).
Die Schrankstützen dürfen während der Nutzung nicht ausgefahren sein.
Der Rechner ist gemäß Hinweisen in der Technischen Beschreibung zu warten (Lüfter, USM, SSM).